COMPUTATION OF HELICOPTER ROTOR WAKE GEOMETRY AND ITS INFLUENCE ON ROTOR HARMONIC AIRLOADS

by

Michael P. Scully

S.B., Massachusetts Institute of Technology
(1964)

S.M., Massachusetts Institute of Technology
(1967)

E.A.A., Massachusetts Institute of Technology
(1967)

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 1975

Signature of Author ____________________
Department of Aeronautics and Astronautics
January 27, 1975

Certified by ____________________
Thesis Supervisor

____
Thesis Supervisor

____
Thesis Supervisor

Accepted by ____________________
Chairman, Departmental Committee on Graduate Students

COMPUTATION OF HELICOPTER ROTOR WAKE GEOMETRY AND ITS INFLUENCE ON ROTOR HARMONIC AIRLOADS

by

Michael P. Scully

Submitted to the Department of Aeronautics and Astronautics on January 27, 1975 in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

ABSTRACT

Efficient techniques are developed for the computation of helicopter rotor tip vortex geometry and helicopter rotor harmonic airloads. These techniques are designed to minimize the computational expense by the use of simplified wake models together with other devices which reduce the amount of integration over the wake of the Biot Savart relation for induced velocity. These techniques are designed for use in steady-state flight at advance ratios below 0.3; however, they could be readily extended to higher advance ratios and to transient flight conditions.

The results of the tip vortex geometry computations show that, over the forward portion of the rotor, the tip vortex is actually much closer to the rotor than the classical rigid wake assumption would indicate. This results in very intense, close blade-tip vortex interactions. These interactions must be modeled much more carefully than the less intense ones which occur when the rigid wake geometry assumption is used. The use of a lifting-surface solution for the airloads induced by these close blade-tip vortex interactions resulted in a substantial improvement in the agreement with measured airloads. Further improvement was obtained by the use of a very simplified model for tip vortex-core bursting and for the propagation of that bursting ahead of the blade which initiated it. This, together with the lifting-surface solution, brought the computed airloads into reasonable agreement with the measured airloads.

A simple tip vortex roll-up model was used to test the effect of roll-up on the airloads computation and it was found to be significant in some cases. The effects of tangentially-induced velocity and local separation on airloads computations were also evaluated and found to be of limited importance. It is recommended that close blade-vortex interactions and tip vortex formation and roll-up be studied both experimentally and theoretically. The currently available knowledge of these phenomena is inadequate for accurate harmonic airloads computations.

Thesis Advisors:

Rene Harcourt Miller, M.A.
Title: H.N. Slater Professor of Flight Transportation,
Head of Department of Aeronautics and Astronautics

Norman Douglas Ham, Sc.D.
Title: Professor of Aeronautics and Astronautics

Eugene Edzards Covert, Sc.D.
Title: Professor of Aeronautics and Astronautics

ACKNOWLEDGMENTS

Grateful acknowledgment is made to Professor R.H. Miller and to Professor N.D. Ham for their advice and inspiration and especially for their friendship. Professor Miller introduced me to this line of research and his continuing advice and supervision has been invaluable. Professor Ham taught me the fundamentals of rotary winged aircraft and has read the manuscript with painstaking care, suggesting many valuable improvements. Professor Ham's family has become my family here on the East coast and there are many happy memories of holidays spent together.

I would also like to thank Professor E.E. Covert and Professor W.S. Lewellen for helpful discussions and advice. Mrs. Gertrude Hubbard deserves especial thanks for her very rapid and efficient typing of this manuscript from a handwritten draft.

This work has been supported by the U.S. Naval Air Systems Command, Airframe Division, largely under contracts N00019-73-C-0378 and N00019-74-C-0321.

TABLE OF CONTENTS

TABLE OF CONTENTS CONTINUED

TABLE OF CONTENTS CONTINUED

TABLE OF CONTENTS CONTINUED

APPENDICES

TABLE OF CONTENTS CONTINUED

TABLE OF CONTENTS CONCLUDED

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS CONTINUED

LIST OF ILLUSTRATIONS CONTINUED

LIST OF ILLUSTRATIONS CONTINUED

LIST OF ILLUSTRATIONS CONCLUDED

LIST OF TABLES

LIST OF SYMBOLS[*]

a	$\frac{\partial C_L}{\partial \alpha}$ before corrections for Mach number and dynamic stall
$\vec{a}, \vec{b}$	Vectors from point P_η to points P_a and P_b on vortex segment (normalized by R) (Fig. 16)
$a_o, a_1, a_2,$ b_1, b_2	Harmonics of rigid blade flapping. $\beta(\psi) = a_o - a_1 \cos\psi - b_1 \sin\psi - a_2 \cos 2\psi - b_2 \sin 2\psi \ldots$
a_R - LDS	Artificial $\partial C_L/\partial\alpha$ including radial flow and dynamic stall effects
$(axb)_i, (axb)_j,$ $(axb)_k$	x,y,z ($\vec{i}, \vec{j}, \vec{k}$) components of $\vec{a}x\vec{b} = (axb)_i \vec{i} + (axb)_j \vec{j} + (axb)_k \vec{k}$
b	Rotor blade semi-chord (ft)
c	Rotor blade average chord (ft)
$\vec{c}$	Vector along vortex line segment from point P_a to point P_b (normalized by R). $\vec{c} = \vec{b} - \vec{a}$ (Figs. A1, B2)
c_a, c_b	Distances along vortex-line segment ($\vec{c}$) from perpendicular dropped from point P_η to $\vec{c}$ to point P_a and P_b (normalized by R) (Fig. B2)
C_T	Thrust coefficient. $C_T = \frac{\text{thrust (lb)}}{\rho \pi R^2 (\Omega R)^2}$
$\vec{D}(\phi, \delta)$ - LDS	Tip vortex distortion at point P_ℓ (normalized by R) (Subsection 1.5)
$\vec{D}(\phi, \delta_\ell)$ - WG	Tip vortex distortion of point P_ℓ (normalized by R)
$\vec{D}(\psi, \delta)$ - WG	Tip vortex distortion at point P_η (normalized by R)

---

[*] The notation (LDS) means that the symbol is used in only the airloads computation or that it has a special definition when used in the airloads computation. The notation (WG) means that the symbol is used in only the wake geometry computation or that it has a special meaning when used in the wake geometry computation. No notation means that the symbol is used in both computations.

$D_1(\phi,\delta)$, $D_2(\phi,\delta)$, $D_3(\phi,\delta)$ - LDS x,y,z components of $\vec{D}(\phi,\delta)$
$\vec{D}(\phi,\delta) = D_1(\phi,\delta)\vec{i} + P_2(\phi,\delta)\vec{j} + D_3(\phi,\delta)\vec{k}$
(normalized by R)

$\vec{D}_\ell(\phi,\delta_\ell)$ - WG Best estimate of distortion at point P_ℓ (normalized by R) (Subsection 4.1.6)

$D_{\ell x}(\phi,\delta_\ell)$, $D_{\ell y}(\phi,\delta_\ell)$, $D_{\ell z}(\phi,\delta_\ell)$ - WG x,y,z components of $\vec{D}_\ell(\phi,\delta_\ell)$
$\vec{D}_\ell(\phi,\delta_\ell) = D_{\ell x}(\phi,\delta_\ell)\vec{i} + D_{\ell y}(\phi,\delta_\ell)\vec{j} + D_{\ell z}(\phi,\delta_\ell)\vec{k}$ (normalized by R)

$\vec{D}_o(\psi)$ - WG Old value of $\vec{D}(\psi,\delta)$ which was replaced by the most recent updated value of $\vec{D}(\psi,\delta)$ in the $\vec{D}$ table (normalized by R)

$\vec{D}_s(\psi)$ - WG Assumed value of $\vec{D}(\psi,\delta_M - \Delta\psi)$ which has been replaced by a new computed value in the $\vec{D}$ table (normalized by R)

$\vec{D}_\eta(\psi,\delta)$ - WG Best estimate of distortion at point P_η (normalized by R) (Subsection 4.1.6)

$D_{\eta x}(\psi,\delta)$, $D_{\eta y}(\psi,\delta)$, $D_{\eta z}(\psi,\delta)$ - WG x,y,z components of $\vec{D}_\eta(\psi,\delta)$
$\vec{D}_\eta(\psi,\delta) = D_{\eta x}(\psi,\delta)\vec{i} + D_{\eta y}(\psi,\delta)\vec{j} + D_{\eta z}(\psi,\delta)\vec{k}$ (normalized by R)

$EI(\bar{\eta})$ - LDS Rotor blade radial distribution of flapwise stiffness (lb-ft^2)

$f_t(\phi,\delta)$ - LDS Fraction of total tip vortex circulation concentrated in tip vortex line at azimuth angle ϕ and age δ.
$f_t(\phi,\delta) = f_t^o(\phi) + \delta * f_t^D$

f_t^D - LDS Tip vortex roll-up rate - $f_t(\phi,\delta)$ increases by $\Delta\phi * f_t^D$ for every $\Delta\phi$ increase in age δ

$f_t^o(\phi)$ - LDS Fraction of total tip vortex circulation concentrated in tip vortex line at azimuth angle ϕ and zero age

F - LDS Theodorsen lift deficiency function

F_c - LDS	Correction factor for tip vortex line segment contribution to induced velocity, accounts for all vortex core and close blade-vortex interaction effects $F_c = q_f * \gamma_c(\rho)$
$G_M(\eta,\psi)$ - LDS	Glauert Mach number correction, $\frac{\partial C_L}{\partial \alpha} = \frac{a}{G_M(\eta,\psi)}$ $G_M(\eta,\psi) = \text{Max}\begin{cases}\sqrt{1-(\eta+\mu\sin\psi)^2 M_T^2}\\ 0.60\end{cases}$
$h, h_1, \vec{h}$	Perpendicular distance for point P_η to a vortex line segment or the forward edge of a vortex sheet segment (normalized by R). (Figs. A1,B2)
i	Angle of incidence of relative wind to TPP (Fig. 15)
$\vec{i}, \vec{j}, \vec{k}$	Unit vectors along x,y,z axes (Fig. 16)
$\vec{i}_{\tilde{x}}, \vec{i}_{\tilde{y}}, \vec{i}_{\tilde{z}}$	Unit vectors along $\tilde{x}$, $\tilde{y}$, $\tilde{z}$ axes (Fig. B1)
I_o, I_1, I_2, I_3 (Subsection 3.3.1) - LDS	Integrals in harmonic blade motion solution (see Eqs. 3.118 to 3.121)
I_1, I_2, I_3, I_4 (Appendix B)	Vortex sheet segment integrals (see Eqs. B.10 to B.13)
I_b - LDS	Rotor blade flapwise moment of inertia about the flapping hinge (slug-ft^2)
ℓ, ℓ_ℓ	Radial station on ℓ blade of point P_ℓ from which an element of vortex wake is shed (normalized by R) (Figs. 5, 16)
$\ell_i(\phi)$ - LDS	Radial station on ℓ blade of outboard edge of inboard trailing vortex sheet (normalized by R) (Subsection 2.5.3)
$\ell_t(\phi)$ - LDS	Radial station on ℓ blade of inboard edge of tip vortex sheet (normalized by R) (Subsection 2.5.3)

$\Delta \ell_{xy}$ - LDS	Distance in TPP from point P_a on the vortex line segment to η blade or its extension, measured parallel to line segment (normalized by segment length). (Fig. 17)
L - LDS	Length of rotor blade outboard of flapping hinge, (ft). $L = (1 - \varepsilon)R$ (Fig. 21)
$L(\eta,\psi)$ - LDS	Airloads per unit radius at radial station η and azimuth angle ψ on the η blade, (lb/in). (Subsection 1.5.1)
LN - LDS	Locke Number (ratio aerodynamic forces/ inertia forces) $LN = \dfrac{2\pi\rho c R^4}{I_b}$
m - LDS	Number of turns of the helical vortex wake used in airloads computation. Maximum age $\delta = 2\pi m$ (Fig. 5)
m - WG	Number of turns of the helical tip vortex whose geometry is computed. Maximum age $\delta = 2\pi m$
$m(\bar{\eta})$ - LDS	Rotor blade running mass (slug/ft)
M - WG	Number of turns of the helical vortex wake below point P_η used in wake geometry computations. Maximum age $\delta_\ell = 2\pi(m + M)$
M_1, M_2, M_4, M_5 M_6, M_7 -LDS	Various mass and stiffness integrals over the rotor blade (Eqs. 3.132-3.137)
M_T - LDS	Rotor blade tip Mach number $M_T = \dfrac{\text{tip speed}}{\text{sound speed}}$
$\vec{n}, \vec{n}_1$	Unit vector parallel to the contribution of a vortex line segment to the vector induced velocity at point P_η (Fig. B1)
n_b	Number of rotor blades

$n_{DM}(\psi)$ - WG	Number of $\Delta\psi$ steps in the boundary age δ_M between successive boundary updates.
n_h - LDS	Number of azimuthal harmonics in harmonic analysis of downwash, airloads and blade motion.
n_ℓ	Number of radial stations ℓ_ℓ on ℓ blade. For LDS-73, $n_\ell = 3$.
n_δ - LDS	$n_\delta = (\frac{\pi}{\Delta\psi} + 1)$ - number of a stations δ used.
n_η	Number of radial stations η_k on η blade
n_ψ	Number of azimuth angles ψ considered. $n_\psi = 2\pi/\Delta\psi + 1$
NFP - LDS	No Feathering Plane - no 1st harmonic feathering (rotor blade pitch rigid body pitching motion) relative to NFP (Fig. 15)
P_a, P_b	Endpoints of the vortex line segment (or of the front edge of the vortex sheet segment) whose contribution to the induced velocity at point P_η is being computed. (Figs. 5, 16) (Subsection 1.5.2)
P_c - LDS	Point on the tip vortex line segment (or its extension) which intersects the η blade (or its extension) when projected into the TPP (Fig. 17)
P_ℓ	Point on the ℓ blade from which an element of vortex wake is shed. (Figs. 5, 16) (Subsection 1.5.2)
P_η - LDS	Point on the η blade at which induced velocities and airloads are computed (Figs. 5, 16) (Subsection 1.5.2)

P_η - WG	Point on the tip vortex at which induced velocities and wake distortion are computed (Fig. 5, 16) (Subsection 1.5.2)
$\Delta\vec{q}$ - WG	Contribution of the last wake segment ($\delta_\ell = 0$ to $\Delta\psi$) to the induced velocity at point P_η (normalized by tip speed)
$\Sigma\vec{q}$ - WG	Total induced velocity at point P_η at age δ less the bound vorticity contribution (normalized by tip speed)
$\vec{q}_{BV}(\psi,\delta)$ - WG	Contribution of the bound vorticity to the induced velocity at point P_η (normalized by tip speed)
$\vec{q}_{CVL}$ - WG	Contribution of a curved vortex line segment to the induced velocity vector at point P_η, (normalized by tip speed)
$\vec{q}_D(\psi)$ - WG	Average induced velocity at point P_η between age $\delta - \Delta\psi$ and age δ (normalized by tip speed)
q_f - LDS	Close blade-vortex interaction factor for tip vortex line induced velocity computations (Subsection 3.2.4)
$\vec{q}_F(\psi,\delta)$ - WG	Contribution of far wake to induced velocity at point P_η (normalized by tip speed)
$q_k(\eta,\psi,\zeta,\ell,\phi)$ - LDS	Contribution of the wake element identified by ζ,ℓ, and ϕ to the z component of the induced velocity at point P_η. (normalized by $\Gamma/4\pi R$)
$\Delta q_k(\eta,\psi,\zeta,\ell,\phi)$ - LDS	Contribution to $q_k(\eta,\psi,\zeta,\ell,\phi)$ of tip vortex sheet segment (normalized by $\Gamma/4\pi R$)
$q'_k(\eta,\psi,\zeta,\ell,\delta)$ - LDS	Same as $q_k(\eta,\psi,\zeta,\ell,\phi)$ but with δ explicit variable instead of ϕ, $\delta = \psi + \zeta - \phi$ (normalized by $\Gamma/4\pi R$)

$\vec{q}_s$ - WG	Contribution to the induced velocity at point P_η of all the wake from one blade with age δ_ℓ (normalized by tip speed)
$q_t, q_t(\eta,\psi,\zeta,\ell,\phi)$ - LDS	Tangential component of the induced velocity contribution of a vortex line (or sheet) segment at point P_η, (normalized by tip speed). $q_t - q_y \sin\psi - q_x \cos\psi$
$q_T, q_T(\eta,\psi)$ -WG	Tangential component of induced velocity (parallel to chord) at point P_η, (normalized by tip speed)
$\vec{q}_{TOT}(\psi)$	If $\Sigma\vec{q}$ is labeled by its azimuth angle ψ and age δ as $\Sigma\vec{q}(\psi,\delta)$, then $\vec{q}_{TOT}(\psi) = \Sigma\vec{q}(\psi,\delta - \Delta\psi)$, (normalized by tip speed), (Fig. 26b)
$\vec{q}_{VL}$	Contribution of a straight vortex line segment to the induced velocity vector at point P_η, (normalized by tip speed)
$\vec{q}_{VS}$	Contribution of a vortex sheet segment to the induced velocity vector at point P_η, (normalized by tip speed)
q_x, q_y, q_z	x, y, z ($\vec{i}$, $\vec{j}$, $\vec{k}$) components of the induced velocity contribution of a vortex line (or sheet) segment at point P_η, (normalized by tip speed)
$q_z(\eta,\psi,\zeta,\ell)$-LDS	z component of the contribution to the induced velocity at point P_η of the wake from radial station ℓ on the ζ blade. (normalized by tip speed)
QMAT(η,ψ,ϕ)-LDS	Influence coefficient for the contribution to the z component of the induced velocity at point P_η of all wake elements at azimuth angle ϕ. (Subsection 3.2.1)
R	Rotor radius (ft)
$S_{1a}, S_{1b}, S_{2a}, S_{2b}$	Expressions appearing in vortex sheet segment integrals. (See Eqs. B.14 - B.17)

TPP	Tip path plane - no 1st harmonic rotor blade rigid body flapping relative to TPP (Fig. 15)
$U_p, U_p(\eta,\psi)$-LDS	Airspeed perpendicular to NFP at point P_η on η blade, (normalized by tip speed). (Fig. 20)
$U_T, U_T(\eta,\psi)$-LDS	Tangential airspeed at radial station η and azimuth angle ψ (point P_η) on η blade, (normalized by tip speed). (Fig. 20)
V_{CH} - LDS	Chordwise velocity in TPP at point P_η (normalized by tip speed)
V_R - LDS	Total (resultant) velocity in TPP at point P_η - vector sum of V_{CH} and V_{SP} (normalized by tip speed)
V_{SP} - LDS	Spanwise (radial) velocity in TPP at point P_η (normalized by tip speed)
V_T - LDS	Rotor blade tip speed (ft/sec). $V_T = \Omega R$
w_f	Weighting factor used in various iterations. New guess = ω_f*(latest estimate + $(1-\omega_f)$* (previous estimate)
x,y,z	Right-hand, rectangular coordinate system with origin at point P_η. Oriented with respect to TPP (Fig. 16)
$\Delta x, \Delta y, \Delta z$	$\Delta x \equiv x_b - x_a$, $\Delta y \equiv y_b - y_a$, $\Delta z \equiv z_b - z_a$ (normalized by R)
$\tilde{x}$, $\tilde{y}$, $\tilde{z}$	Right-handed, rectangular coordinate system with origin at point P_η. Oriented with respect to each individual vortex-sheet segment. (Fig. B1)
$\Delta\tilde{x}$	Length of a vortex sheet segment perpendicular to its circulation vector (normalized by R) (Fig. B3)
x_1, x_2	$\tilde{x}$ coordinates of forward and back edges, respectively, of vortex-sheet segment (normalized by R). (Fig. B1)

$x_a, y_a, z_a,$ x_b, y_b, z_b	x,y,z components of $\vec{a}$ and $\vec{b}$ extending from point P_η on the blade to endpoints P_a and P_b of a vortex line segment (or of the front edge of a vortex sheet segment) (normalized by R). (Fig. 16)
x_h, y_h, z_h	The x,y,z components of $\vec{h}$. $\vec{h} = x_h\vec{i} + y_h\vec{j} + z_h\vec{k}$ (Appendix B)
x_x, y_x, z_x	The x,y,z components of $\vec{i}_{\tilde{x}}$ $\vec{i}_{\tilde{x}} = x_x\vec{i} + y_x\vec{j} + z_x\vec{k}$ (Appendix B)
x_y, y_y, z_y	The x,y,z components of $\vec{i}_{\tilde{y}}$ $\vec{i}_{\tilde{y}} = x_y\vec{i} + y_y\vec{j} + z_y\vec{k}$ (Appendix B)
$z(\eta,\psi)$ - LDS	Vertical displacement of point P_η above NFP (normalized by L)
z_c - LDS	Distance of vortex-line segment or its extension above or below η blade (normalized by R) (Fig. 18)
z_h - LDS	z component of distance from point P_η to nearest point on vortex-line segment (normalized by R)
z_m	In vortex sheet computations the mean z component of vectors from point P_η to the endpoints of the back edge of the vortex sheet (normalized by R) (Fig. B3)
z - LDS	Perpendicular distance from point P_η to the vortex sheet segment (normalized by R) (Fig. B1)
$\alpha, \beta, \alpha_1, \beta_1$	Angles between the vectors $\vec{a}$ and $\vec{b}$ (from point P_η to point P_a and P_b at the ends of a vortex line segment) and the vortex line segment. (Fig. A1)

$\alpha(\eta,\psi)$	Angle of attack at point P_η - radial station η and azimuth angle ψ. (Fig. 20)
$\dot{\alpha},\dot{\alpha}(\eta,\psi)$ - LDS	Time rate of change of angle of attack, $\dot{\alpha} = \partial\alpha/\partial\psi$
α_R - LDS	Reference angle of attack for dynamic stall (radians)
α_s - LDS	Input stall angle of attack (radians, but input in degrees)
α_{ss} - LDS	Computed statis stall angle of attack (radians), Eq. 3.223 or Eq. 3.224
γ_o	Constant value of bound circulation used in zeroth iteration (normalized by $2\pi b\Omega R$). Normally, $\gamma_o = 2(C_T/\sigma)/\pi(.8)^2$
γ_a	Vortex line (or sheet) segment circulation for constant circulation segments (normalized by $2\pi b\Omega R$)
γ_a,γ_b	Vortex line (or sheet) circulation at points P_a and P_b, respectively, for segment having linear circulation variation (normalized by $2\pi b\Omega R$)
$\gamma_{bv}(\psi)$, $\gamma_{bv}(\phi)$	Bound circulation as a function of azimuth (normalized by $2\pi b\Omega R$)
$\gamma_c(\rho)$, $\gamma(\rho)$	Fraction of total tip vortex line circulation within ρ of the center of the line (Subsection 2.4)
$\gamma_n(\eta,\psi)$ - LDS	Bound circulation from the n'th iteration at point P_η - radial station η and azimuth angle ψ, (normalized by $2\pi b\Omega R$)
$\gamma_s(\psi),\gamma_s(\phi)$	Shed circulation as a function of azimuth (normalized by $2\pi b\Omega R$) $\gamma_s(\psi) = \gamma_{bv}(\psi + \Delta\psi) - \gamma_{bv}(\psi)$
Γ	Vortex line (or sheet) segment circulation (ft^2/sec)

δ - LDS	Age of point P_ℓ on the wake element which was shed from station ℓ on the ℓ blade (radians) (Subsection 1.5.2)
δ - WG	Age of point P_η on the tip vortex where $\vec{D}(\psi,\delta)$ is computed, measured in azimuth angle (radians) (Subsection 1.5.2)
$\delta_b(\phi)$	Age at which the tip vortex line segment at azimuth angle ϕ will experience vortex core bursting (radians) (Subsection 2.5.1)
δ_D - WG	Value of δ for the most recent computation of $\vec{D}(\psi,\delta)$ (radians)
δ_ℓ - WG	Age of point P_ℓ on the wake element which was shed from station ℓ on the ℓ blade (radians) (Subsection 1.5.2)
δ_M - WG	Maximum value of age δ for which the distortion $\vec{D}(\psi,\delta)$ is being computed, thus the boundary between computed and assumed wake (radians)
δ_R - WG	Relative age of point P_ℓ measured relative to point P_η instead of relative to the rotor (radians) $\delta_R = \delta_\ell - \delta$
δ_{ST} - WG	Initial value of δ_ℓ (Fig. 26c)
$\delta_T(\zeta,\psi,k)$ - WG	Value of relative age δ_R at which a far wake/near wake or near wake/far wake transition occurs (radians)
ε - LDS	Rotor blade flapping hinge offset (normalized by R)
ε_s - LDS	Distance behind η blade to start of near shed wake vortex sheet (normalized by R) (Subsection 2.7)
ζ, ζ_i	Azimuth angle difference between ith ℓ blade and η blade. For n_b bladed

	rotor: $\zeta_i = (i - 1)(2\pi/n_b)$ (Figs. 5, 16) (Subsection 1.5.2)
η, η_k	Radial station on η blade of point P_η at which induced velocity is computed. η_1 = inboard, η_{n_η} = outboard ($k = 1, n_\eta$) (normalized by R) (Figs. 5, 16) (Subsection 1.5.2)
$\overline{\eta}$ - LDS	Radial distance from flapping hinge (normalized by L), $\overline{\eta} = (\eta - \varepsilon)/(1 - \varepsilon)$ (Fig. 21)
$\Delta\eta$ - LDS	Radial distance from point P_η to point P_c on blade. $\Delta\eta = \eta_c - \eta$ (Fig. 17) (normalized by R)
η_c - LDS	Radial location of passage of vortex line segment (or its extension) above or below η blade (or its extension) (normalized by R) (Fig. 17)
$\Delta\eta_k$	Radial increment used in radial integrations - multiplies value of integrand at η_k (normalized by R)
$\theta(\eta)$ - LDS	Rotor blade pitch angle relative to NFP (no feathering plane) at radial station η (radians) (Fig. 20)
θ_o - LDS	Rotor blade collective pitch at zero radius (extrapolated) relative to NFP (radians)
θ_1 - LDS	Rotor blade linear twist: $\theta(\eta) = \theta_o + \theta,\eta$
λ	Uniform, average downwash perpendicular to TPP - used to determine rigid wake geometry (normalized by tip speed)

$\lambda(\psi,\eta)$ - LDS	Computed downwash perpendicular to TPP at point P_η - at radial station η and azimuth ψ, (normalized by tip speed)
$\lambda_o(\eta), \lambda_{nc}(\eta), \lambda_{ns}(\eta)$ - LDS	Azimuth harmonics of $\lambda(\psi,\eta)$: $\lambda(\psi,\eta) = \lambda_o(\eta) + \lambda_{1c}(\eta)\cos\psi + \lambda_{1s}(\eta)\sin\psi + \ldots + \lambda_{nc}(\eta)\cos n\psi + \lambda_{ns}(\eta)\sin n\psi + \ldots$
μ	Advance ratio ($V/\Omega R$), V = flight speed (relative wind), ΩR = tip speed
ρ	Air density (slug/ft^3) - also: radial coordinate from the center of a vortex line (normalized by vortex-core radius ρ_c)
ρ_c	Radius of the vortex core (normalized by R)
ρ_g	Burst value of ρ_c - vortex-core radius (normalized by R)
ρ_k	Nonburst value of ρ_c - vortex core radius (normalized by R)
σ	Rotor solidity: $\sigma = \frac{\text{rotor blade area}}{\text{rotor disk area}}$ $\sigma = \frac{cn_b}{\pi R}$
ϕ	Azimuth angle of ℓ blade, measured from same reference as ψ $\phi = \psi + \zeta - \delta$ (see Figs. 5, 16) (Subsection 1.5.2)
$\Delta\phi$	Azimuth angle increment between successive positions of the ℓ blade. (Aximuthal extent of trailing vortex-line segment).
ψ, ψ_j	Azimuth angle of η blade, $\psi = 0$ is downstream, $\psi_j = (j-1)\Delta\psi$, $(0 \leq \psi_j \leq 2\pi)$ (Figs. 5, 16) (Subsection 1.5.2)

$\Delta\psi$	Azimuth angle increment between successive positions of the η blade
Ω	Rotor angular velocity (radian/sec)

SECTION 1

INTRODUCTION

1.1 Importance of Wake Distortion and Harmonic Airloads

Rotary winged aircraft suffer from excessive vibration. This vibration makes passengers uncomfortable, reduces crew efficiency, and induces fatigue failure in both structure and equipment. Some helicopters are at present vibration limited rather than power limited in their maximum usable speed capability. All rotary winged aircraft suffer from increased maintenance costs and reduced passenger appeal due to this excessive vibration. It is therefore important to understand and be able to predict the primary source of vibration: rotor harmonic airloads.

Rotor harmonic airloads come primarily from two sources, the once per revolution variation in velocity tangential to the blade,due to forward flight, and the rapid variations in velocity perpendicular to the blade, due to the vortex wake. Classical uniform inflow theory includes the first source of harmonic airloads, which is relatively easy to compute,but not the second,which is much more difficult to compute. Unfortunately, nonuniform inflow is the primary source of high harmonic airloads and hence vibration, so efficient nonuniform inflow computation techniques must be developed.

A complete model for the computation of the harmonic airloads, due to the nonuniform inflow induced by the vortex wake, would include: real (viscous) vortices, lifting-surface theory, compressibility, dynamic stall, and a detailed wake model with an accurate wake geometry, including the roll-up of part of the trailing vortex sheet into the tip vortex. An all-encompassing computation using such a model is impractical (i.e., it would cost too much). It is, therefore, necessary to develop an approximate model which will reduce computation to a practical level.

The first step is to develop a better physical understanding of the various sources of harmonic airloads and their relative importance. It then becomes possible to use a very simple model for the bulk of the computation and transition to more complicated models for those areas of the wake which prove to be the most important sources of higher harmonic airloads. When constructing these models it is important to realize that most of the expense in a harmonic airloads computation is in the integration of the Biot-Savart relation over the wake to find the induced velocities at various points both in the wake and at the rotor blades. The best way to save money is, therefore, to minimize the number of elements in the wake model, the number of points at which the induced velocity is computed, and the number of times that the induced velocity at a given point due to a given wake element is recomputed.

The dominant source of rotor harmonic airloads is the rapid variation of the induced velocity at the rotor blades, both radially and azimuthally. This rapid variation is primarily due to the passage of the rotor blades close to concentrated vortices in the wake. The vortex wake is composed of the trailing wake, due to radial circulation variations, and the shed wake, due to azimuthal circulation variations. The only concentrated vortices in the wake are the tip vortices trailed from the tip of each blade. These tip vortices are due to the very rapid drop in circulation near the blade tip. Most of the trailing wake outboard of the maximum bound circulation rapidly rolls up into the concentrated tip vortex. The rest of the trailing wake and the shed wake does not roll up until it is far downstream of the rotor, if at all. The tip vortices are, therefore, the most important part of the wake and the primary source of harmonic airloads.

To compute rotor harmonic airloads, the harmonic induced velocity at the rotor blades is computed by integration of the Biot-Savart relation over the vortex wake. To perform this integration, the location of the wake relative to the blades must be known. The location of the inboard trailing wake and the shed wake need not be known to great accuracy, since modest errors in their location do not change the induced velocity distribution very much. Due to the concentrated nature of the tip vortices, errors in their location can lead to large errors in the induced velocity distribution and hence in the harmonic airloads. Therefore, a "rigid" wake geometry based on uniform inflow is adequate for the inboard trailing wake and the shed wake, but a more accurate "distorted" wake geometry should be used for the tip vortices. The distortion is due to nonuniformities in the inflow and is computed by integration of the sum of the free stream and induced velocities over time at various points on the tip vortices.

Figure 1 is a comparison of rigid and distorted tip vortex geometries. The data is for the tip vortex of one blade of a four-bladed rotor in steady, level flight at an advance ratio, $\mu = 0.18$. The figure shows the geometry at the instant when the rotor blade is at an azimuth angle $\psi = 180°$; however, the geometry was computed for all azimuth angles ψ between 0° and 360° in 15° steps.

Figure 2 is a comparison of airloads computed using both rigid and distorted tip vortex geometry. Only the airloads at the 95% radial station are shown; however, 5 other radial stations were computed. The bound circulation distribution used for the tip vortex geometry computation of Fig. 1 comes from this rigid wake airloads computation. The tip vortex geometry used in the distorted wake airloads computation of Fig. 2 comes from the wake geometry computation of Fig. 1. Measured airloads from the flight tests reported in [49]* are also shown in Fig. 2.

---

*References are indicated by a number in square brackets [ ].

As can be seen in Fig. 1, there are substantial differences between rigid and distorted tip vortex geometries. This is especially true in the side view,where the distorted tip vortex is seen to stay in the tip path plane (TPP) until the next rotor blade arrives. This would be expected to cause dramatic changes in airloads, yet Fig. 2 shows relatively small changes. This surprising result is due to various close blade-vortex interaction effects,which tend to smooth out large airloads peaks. If these effects are neglected, there can be as much as an order of magnitude difference in airloads peak heights between rigid and distorted wake computations. These close blade-vortex interaction effects include: lifting-surface effects, tip-vortex-core effects, tip vortex core bursting, and local stall effects.

1.2 Previous Investigations

The ability to accurately compute helicopter rotor harmonic airloads in forward flight has steadily advanced over the last 25 years, due to the increased problems of modern, high-speed helicopters and due to the availabiltiy of high-speed digital computers. The first step beyond the classical assumption of uniform inflow was the development of theories assuming an infinite number of blades, for example [1-4]. These theories are simple enough to be used in hand computations or in tabulated solutions. These theories distribute the tip vortices from the individual blades into a single vortex cylinder leaving no concentrated tip vortices in the wake model. Therefore, such theories cannot predict high harmonic airloads,which mostly arise from close interactions between the rotor blades and concentrated tip vortices.

The next step is the utilization of high-speed digital computers. This approach permits the representation of a finite number of rotor blades where the wake of each blade is represented separately. The wake of each blade is assumed to consist of a series of vortex lines,each forming a skewed helix whose geometry is determined by the forward speed of the helicopter (advance ratio) and the mean inflow through the tip path plane. This is the rigid wake assumption. The first of these computer solutions uses numerical integration of the Biot-Savart relation over the skewed helices [5-8]. This procedure requires a very small step size to achieve adequate accuracy. The use of infinite straight vortex lines tangent to the skewed helices at carefully selected points is found in [9,10]. This technique minimizes computation, since the Biot-Savart relation can be integrated in closed form, but accuracy is sacrificed due to neglect of the curvature of the wake. The best solution to this problem to date is the approximation of the skewed helices by a series of finite length straight vortex line segments, each subtending the same azimuth angle [10-16]. The Biot-Savart relation can be integrated in closed form over a finite straight vortex line segment. Large step sizes can, therefore, be used compared to direct numerical integration, where large step sizes result in large variations in the computed induced velocity depending on the choice of points at which the integrand is computed.

This rigid wake analysis gives a good first approximation to the induced velocity and hence the airloads due to the trailing wake. It is also necessary to include the effects of azimuthal (time) variations in circulation and the resulting shed wake. One way to do this is to use a constant average circulation in the trailing wake computations and lump all circulation variation effects into a Theodorsen lift-deficiency function [5-9]. This only accounts for circulation variations in the near wake,since it is based on two-dimensional wing theory. The far shed wake can become important when it passes under another blade or returns to pass under the original blade. To account for circulation variations in the far wake, each trailing vortex line segment is assigned a different circulation and shed vortex line segments are introduced as necessary to conserve circulation [9,14]. Instead of using a lift-deficiency function to account for near wake effects, it is often computationally convenient to continue the integration of the shed wake up to within ε of the control point on the blade, where ε is determined in [9] by the condition that the result be consistent with classical theory.

As was discussed in Section 1.1, the exact location of the tip vortex relative to the blades is important in the computation of harmonic airloads. The first step beyond the rigid wake assumption is the semi-rigid wake [7,17]. This model allows each wake element to move downward (perpendicular to the tip-path plane) with its actual initial local inflow velocity at the instant it leaves the blade, instead of an average inflow velocity which is uniform over the disk. This model is better than rigid wake,but not completely satisfactory since the induced velocity experienced by any wake element changes as it moves and may even change sign. In addition,significant distortion of the wake can occur parallel to the tip-path plane as well as perpendicular to it.

The solution to these problems is to utilize a distorted (or free) wake computation in which the induced velocity vector is computed as a function of time for various wake elements and integrated to obtain the distortion vector. In a wake distortion computation, the current best estimate of the wake geometry changes as the computation proceeds. This means that both the location of the wake elements where the induced velocity vector is being computed and the geometry of the wake which induces the velocity is changing. It is, therefore, necessary to recompute or update repeatedly the integration of the Biot-Savart relation over the wake to obtain the induced velocity vectors. This is very expensive.

To reduce the amount of expensive computation, it is desirable to minimize the amount of updating. One starting point is to use an iterative scheme based on the assumption that the distorted wake geometry will not be very different from the classical rigid wake assumption [18]. This scheme involves starting with a segment of rigid wake $2\pi m$ in azimuth (typically, $m = 2$). The rotor is then allowed to advance, generating new wake, and the distortion of this new wake is computed based on the $2\pi m$ of the rigid wake plus the new distorted wake. No updating is used and when $2\pi M$ ($M \leq m$) of new wake has been generated, one iteration is completed. The next iteration starts using the $2\pi M$ of distorted wake from the previous iteration,plus any extrapolation needed to fill out the $2\pi m$ segment. This iterative process continues until satisfactory convergence is achieved. This method is cheap, since no updating is done, but it converges very slowly or not at all and it is not very accurate. This is because small changes in wake geometry can result in very large changes in the induced velocity vector for points which happen to lie near a concentrated vortex line.

To improve this situation,some updating of the induced velocity vectors must be introduced. The first step in this direction is to update the induced velocity vectors after every nth step in the computation [19]. This greatly improves both convergence and accuracy but is expensive, especially for small n. An alternative to updating every nth step is to update any given azimuth, whenever the distortion accumulated at that azimuth since it was last updated exceeds a predetermined value. This has been found to be less satisfactory than updating every n steps because it turns out that the time spent accumulating distortion due to an inaccurate induced velocity vector is more important than the amount of distortion accumulated. What sometimes happens, is that a very small amount of distortion at some azimuth, combined with the distortion of the wake inducing velocity at that azimuth, can result in a 180° phase change in the induced velocity vector. This phase change is not found until enough distortion is accumulated in the wrong direction to trigger a new update.

An alternative approach to computing wake geometry has been developed concurrently with the one described above [20,21]. This involves updating after every step both the induced velocity vectors and the distortion vectors for all of the wake whose distortion has been computed to date. These computations may start with either a segment $2\pi m$ of rigid wake or with no wake at all. They continue generating new wake until a stable distortion pattern is achieved. Unfortunately, this straightforward method is very expensive, due to the large number of times that the induced velocities must be recomputed.

A major improvement over the above procedure is developed in [22]. During computation of the induced velocity at any point P_η on the tip vortex, the rest of the wake is divided into two parts relative to point P_η, the near wake and the far wake. The induced velocity at point P_η due to the entire wake is computed for some assumed wake geometry such as a "rigid wake". The near wake is then composed of those wake elements (straight vortex line segments) which induce at point P_η a velocity greater than some input value q_m. The rest of the wake becomes the far wake relative to point P_η. Naturally, each different point P_η on the tip vortex will have a different near and far wake. The induced velocity at point P_η due to the far wake is stored. Whenever the induced velocity at point P_η is updated, only the near wake contribution is recomputed, the stored value is used for the far wake. The value of q_m is a trade-off between computation time and accuracy and its selection is based on experience. This near wake technique results in about an order of magnitude reduction in computation time, which is a great improvement. There is a price for this reduction in computation time, and that is a major increase in the requirement for fast access memory capacity (core storage). This may force the use of very large interval sizes or even prohibit the use of the near wake technique, depending on the computer available.

The developments described above are primarily for the forward flight case. Hover is very different from forward flight because in hover there is no large uniform relative wind. This means that the wake does not spread out as rapidly and hence more of the wake must be considered when computing either airloads or wake geometry. It also means that induced velocities are the only velocities present and therefore wake geometry is much more sensitive to changes in induced velocity. This increased sensitivity leads to instabilities in the tip vortex geometry as shown in [23,24].

Classical propeller theory [25,26] assumes no slipstream contraction. This is adequate for axial flight,but not statically. One solution to this problem is to determine the slipstream contraction envelope based on actuator disk theory. This envelope can be used directly in an actuator disk solution for airloads [27] or as the outer boundary of the vortex sheets trailed by each rotor blade [28]. To date these envelope solutions have not produced completely satisfactory results.

Instead of the envelope approach, the distortion of the wake of each blade can be computed directly in a manner similar to the forward flight wake geometry computation methods of [19, 22]. These methods [23,29,30] still do not accuately predict the tip vortex geometry near the rotor. In the far wake, the method of [23,29] predicts instability which agrees with experimental data [23,31]. The method of [30] does not predict this instability but instead computes an average location of the wake. This is due to computing induced velocities at fixed points in space rather than at points on the wake.

Since none of the theoretical methods of computing hovering rotor wake geometry is yet completely satisfactory, an empirical correlation method has been developed based on model test data [23,29,31,32]. This correlation method is currently the most useful method of computing hovering rotor wake geometry. It is limited to relatively conventional rotors at present and a better theoretical model is still desirable.

This quick survey of previous investigations is intended to place the curren work in the context of previous and parallel efforts in this field. A more general survey of the literature can be found in [33].

1.3 Current Investigation

The concentrated tip vortices are the primary source of vibratory rotor airloads. It is, therefore, necessary to be able to compute tip vortex geometry as a part of an accurate vibratory airloads computation. Methods have been developed for computing tip vortex geometry, but they are very expensive. The current investigation therefore concentrates on developing inexpensive tip vortex geometry computation techniques with provisions for trading-off accuracy for cost. In addition, an efficient airloads computation is developed, both to provide the circulation distribution as an input to the tip vortex geometry computation and to determine the effect of tip vortex geometry on harmonic airloads. The special case of hovering rotor tip vortex geometry is also studied.

1.3.1 Wake Models

The first step is the development of the simplest mathematical model for the vortex wake which gives acceptable results. The wake is divided into three parts: the tip vortex, the inboard trailing wake, and the shed wake. Models are developed for each of these parts. These models are made up of straight vortex line segments, curved vortex line segments, and plane rectangular vortex sheet segments. These segments can either be of uniform circulation strength or they can have a linear variation in circulation strength. The vortex line segments have a vortex core in which viscous effects become important. In some cases, vortex line segments having an artificially large vortex core are used to represent vortex sheet segments at a considerable saving in computation time.

A basic wake model is developed. The tip vortex is represented by a vortex line made up of straight vortex line segments. Curved vortex line segments are used for those segments of tip vortex, adjacent to the point P_η on the tip vortex, at which the induced velocity is being calculated during a wake geometry computation. The inboard trailing wake is represented by either a vortex sheet or by a vortex line having an artificially large vortex core. The shed wake can be represented by either a series of vortex sheet segments or by a series of vortex line segments having artificially large vortex cores. In any case, the near shed wake is represented by a vortex sheet segment extending to within a distance ε of the control points on the rotor blade. This basic wake model and various refinements of it are discussed more fully in Section 2.

Due to the lack of relative wind, the wake models used in the computation of hovering tip vortex geometry are somewhat different from those for forward flight. This is discussed further in Section 1.3.5.

1.3.2 Circulation-Wake Geometry Iteration

For the forward flight case, two separate computer programs have been developed: one for computing tip vortex geometry and one for computing rotor airloads. The airloads program (LDS) uses a given tip vortex geometry which is held constant for the duration of the computation. This geometry can be either a "rigid wake" geometry based on a uniform inflow or a distorted geometry computed by the tip vortex geometry program. The tip vortex geometry program (WG) uses a circulation distribution which remains unchanged for the duration of the computation. This circulation distribution is normally obtained from a previous run of the airloads program. As shown in Fig. 3, these two programs are normally run together in a circulation distribution, tip-vortex geometry iteration. The first step in this iteration is usually an airloads computation using the "rigid wake" tip vortex geometry. Tip-vortex geometry is generally not very sensitive to moderate variations in the circulation distribution. Thus one iteration involving an initial "rigid wake" airloads computation, a wake geometry computation, and a final "distorted wake" airloads computation is usually sufficient.

1.3.3 Tip Vortex Geometry Computation

The method of computing tip vortex geometry is an extension and refinement of the method developed in [19]. The modifications fall into two categories: those which increase accuracy at the expense of some increase in computation time and those which greatly reduce the computation required with a minimal sacrifice in accuracy. The net result is a method which uses the basic iterative approach of [19], but has been completely reworked to achieve both increased accuracy and a reduction in the amount of computation required.

The major saving in computation comes from the adoption of a near-wake/far-wake scheme similar to that used in [22]. Such a scheme was considered during the development of the method reported in [19]. The scheme was rejected because it required more fast access computer memory (core storage) than was available at M.I.T. at that time. The publication of [22] inspired a second look at the problem to find a way of reducing the core storage requirements to a level practical for use on the available computer. The result is a near-wake/far-wake scheme which achieves most of the savings in computation obtained in [22] with a much more modest core storage requirement. This scheme is described in Section 4.

In [19] only the induced velocities at points on the boundary between the "new" wake,whose geometry has been computed during the current iteration,and the "old" wake,whose geometry has not been computed yet,are updated. This form of updating has been retained,with the refinement that the frequency of updating is a function of azimuth. This allows more frequent updating to be specified at the sides,where many different tip vortices interact closely,compared to fore and aft,where the tip vortices are more spread out.

In addition, a second, more complete type of updating has been introduced. In this case, induced velocities are updated for all of the "new" wake, not just the points on the boundary. This type of updating requires computation of the induced velocity at many more points than the first type of updating and it is done much less frequently. The use of this second type of updating requires a substantial increase in the amount of computation. On the other hand, it improves the accuracy and stability of the wake geometry computation, reducing the number of iterations required and improving the accuracy of the results. Fortunately, the saving in computation achieved by using the near-wake scheme more then compensates for the extra computation required by the new updating process.

In addition to the major modifications to the methods of [19] described above, various refinements have been introduced. Instead of basing the wake geometry on only the most recent computation of the distortion, at each stage in the computation the latest computation of the incremental distortion during the latest step in age (time) is combined with the immediately preceding computation of the same distortion in a weighted average. This averaging greatly improves the stability and the convergence of the tip-vortex geometry computation. Another refinement is the use of closed-form integration of the induced velocity over time to compute the distortion due to the bound vorticity when a rotor blade passes over an element of the tip vortex. This avoids the problem of sampling the induced velocity due to the bound vorticity at discrete points in an area of rapid variation.

In summary, a new tip-vortex geometry-computation method has been developed starting from the method of [19]. The major developments are:

(a) The use of more sophisticated wake models using vortex sheets, vortex lines having artificially enlarge vortex cores, and vortex core bursting.
(b) The near-wake/far-wake scheme.
(c) Modification of the original type of updating and the addition of a new type of updating.

In addition, various lesser developments have been introduced. For a more detailed account of these developments, see Section 4.

1.3.4 Harmonic Airloads Computation

As shown in Section 1.3.2, an airloads computation is needed to supply the circulation distribution to the wake geometry computation. It must also be remembered that the actual goal is the improved computation of rotor harmonic airloads, not just the computation of tip-vortex geometry. Thus an airloads computation is necessary to evaluate the effects of tip-vortex geometry on rotor airloads.

To these ends an airloads computation has been developed. It is described in detail in Section 3, so only its more important or unusual features will be outlined here.

1) The wake models used are generally similar to those used in wake-geometry computations.
2) There is provision for a jump in tip-vortex-core size after a close blade-vortex interaction.
3) The "rigid-Wake" geometry is always used for the inboard portion of the wake and can be used for the tip vortex. Alternatively, the distorted tip-vortex geometry obtained from a wake-geometry computation can be used.

4) Normally, lifting-line theory is used but for close blade-tip-vortex interactions, lifting-surface theory is used.
5) Radial flow, dynamic stall, and some compressibility effects are represented by simplified models.
6) Rigid-body flapping and first mode blade bending (with an assumed mode shape) are included in the blade-motion solution. No lag or torsional motion of the blade is included. Flapping hinge offset is included and a spring restraint about the flapping hinge can be used to represent "rigid" rotors.

The computation of airloads with distorted tip-vortex geometries, using the above techniques, demonstrated that the methods for computing close blade-tip vortex interactions are inadequate. This lead to experiments with various models of the interaction process. These experiments have lead to some rather crude, empirical models which give somewhat better results. They also demonstrate the critical importance of close blade-tip vortex interactions in determining the harmonic airloads and the necessity for basic research in this area. Until a better model for close blade-tip vortex interactions is available, the computation of accurate tip-vortex geometries, at considerable expense, is hard to justify, except in special cases such as hover.

1.3.5 Hovering Wake Geometry

The hovering tip-vortex geometry computation is developed from the general forward-flight computation. Axial symmetry of the wake geometry is assumed (i.e., any wake element having a given radial station η and age δ will have the same geometry for all azimuth ψ). This symmetry assumption greatly simplifies the computation ,since tip-vortex geometry and hence induced-velocities are calculated at only one azimuth instead of at twenty-four azimuths.

Since there is no relative wind in the hovering case, the wake does not move away from the rotor as rapidly as in the forward flight case. This requires that a hovering tip-vortex-geometry computation consider more of the wake than a forward flight computation. Typically 12π (6 turns) of hovering tip vortex geometry is computed and as much as 24π (12 turns) of wake is considered below point P_η when computing the induced velocity at that point. In addition, a semi-infinite vortex cylinder is used to represent the tip vortex from the end of the normal wake model to infinity. Besides the normal models used to represent the inboard trailing wake, there is provision for the use of tilted vortex sheets based on the experimental observations of [23].

A specialized airloads computation for the hovering case has not been developed. Due to the assumption of symmetry, the circulation is constant around the azimuth. The tip vortex circulation used in tip vortex geometry computations is obtained either by simple uniform inflow computations or from the experimental data. This is adequate for current use but an operational hovering tip-vortex geometry computation would require a corresponding airloads computation.

The results obtained from the hovering tip-vortex geometry computation near the rotor are similar to those obtained by other investigators [23,30] making similar computations (i.e., the theory does not predict rapid enough wake contraction or downward movement near the rotor). In the far wake, convergence to a stable tip-vortex geometry is achieved only by the introduction of weighted averaging between iterations as described in Section 1.3.3. This convergence may be artificial and only represents an average position of the tip vortex, since there is some evidence that the far wake is unstable in reality [23,24].

1.4 Assumptions and Approximations

This section is a review of the basic assumptions and approximations used in the computation of tip-vortex geometry and harmonic airloads. The basic assumption of steady state, equilibrium flight is made. This means that the rotor angular velocity Ω, the helicopter flight speed (or the relative wind speed) V, the incidence of the tip-path plane relative to the wind, i, and the net thrust and torque of the rotor all remain constant over time. It is further assumed that all rotor-blade referenced quantities such as airloads, bound circulation, and blade motion are cyclic in azimuth angle every 360°. This implies that the wake circulation is also cyclic every 360° and that the wake geometry is stable.

A simplified basic model is used to save computation. This basic model can be described as unsteady, three-dimensional, inviscid and incompressible lifting-line theory. This basic model is satisfactory for the bulk of the computations, but is is modified in some regions. Near a concentrated vortex line, such as the tip vortex, a special vortex-core model is used. Viscous effects such as stall and tip vortex core bursting are also modeled. Compressibility effects on lift-curve slope, on maximum lift coefficient, and on drag are included. For close rotor-blade/tip-vortex interactions, lifting-surface theory is used. Thus the more important effects of viscosity, compressibility, and lifting-surface theory are included without the expense and complication of using a complete viscous and compressible lifting-surface theory.

A set of formulae is used to approximate airfoil section lift and drag as a function of Mach number, angle of attack and time rate of change of angle of attack. These formulae are typical of modern cambered helicopter airfoils, but do not represent any particular airfoil. Dynamic stall, radial flow, and reverse flow effects are modeled. The tangential component of induced velocity can be included if desired, but it is not normally very important.

In the blade-motion solution, rigid-body flapping and first mode flapwise blade bending are included, but lag and torsional motion are not. The effects of flapping hinge offset are included but lag and torsional motion are not. The effects of flapping hinge offset are included and a spring about the flapping hinge can be used to simulate a cantilever rotor. In both the calculation of blade motion and of harmonic airloads, all harmonics of blade motion are included and no small angle assumptions are made. In the determination of the relative positions of the wake and the rotor blades, however, the zeroth and first harmonics of flapping are assumed to be small angles and the higher harmonics of flapping plus all harmonics of blade bending are neglected.

1.5 Nomenclature and Notation

1.5.1 Fundamental Nomenclature

This section defines the most important nomenclature used in this report. Less important, or less widely used nomenclature will be defined where it is first used. To start with fundamentals, consider the words: airloads (or harmonic airloads), wake geometry, and distortion.

Airloads are the rotor blade section aerodynamic lift per unit radius (lb/in). The lift is that component of the total section aerodynamic force which is perpendicular to the local relative wind (neglecting radial flow). Airloads vary with both the rotor blade azimuth angle (ψ) and the radial station (η) on the rotor blade. The term harmonic airloads is used to emphasize that all azimuthal harmonics of airloads are of interest here, not just the steady-state component.

Wake geometry is the geometric relationship between the various elements of the wake and of the rotor. All wake geometry is measured relative to the rotor tip-path plane (TPP). This eliminates the effect of first harmonic flapping on wake geometry. The effects of higher harmonics of flapping on wake geometry are neglected.

The wake geometry is divided into two parts: the fundamental wake geometry and the distortion. The fundamental wake geometry is that due to the motion of the rotor blades relative to the free stream (neglecting higher harmonic flapping). Thus the fundamental tip-vortex geometry is the skewed helix described by the rotor blade tip as it moves relative to the free stream. The distortion is that part of the wake geometry which is due to the velocities induced by the wake and by the bound circulation. In the "rigid wake" assumption, the induced velocity is uniform and perpendicular to the TPP. Thus, for this case, the tip vortex distortion is obtained by adding the uniform induced inflow to that component of the free stream which is perpendicular to the TPP. This results in the stretching of the fundamental skewed helix along an axis perpendicular to the TPP.

In the general distorted wake case, the induced velocity can be nonuniform, time varying, and three dimensional. Consider the location of a general point P_ℓ in the wake. Point P_ℓ will have a position (P_ℓ^F) on the fundamental skewed helix which remains fixed relative to the air in the free stream. The distortion vector ($\vec{D}$), drawn from P_ℓ^F to the actual location of point P_ℓ, will, in general, vary with time and with the choice of point P_ℓ.

1.5.2 Fundamental Notation

This section defines the basic notation needed to follow the more general portions of this report. Other, less important notation is defined as it first occurs in the text. Most of the notation used here is defined in an alphabetical list of notation at the beginning of this report. Any notation not defined in this list is very specialized and is only used in one small part of the report, such as an appendix.

Point P_η is a general point at which the induced velocity is being computed. For airloads computations, P_η is a point on the η blade of the rotor. The η blade has an azimuth angle ψ and point P_η is at radial station η. Radial station η is a distance ηR from the center of rotation, where R is the rotor radius.

For wake geometry computations, P_η is a point in the wake. The element of wake at point P_η was originally shed from radial station η on the η blade, which had an azimuth angle ψ at the instant of shedding. The current age δ of the element of wake at point P_η is the difference between ψ and the current azimuth angle of the η blade. Thus the azimuth angle, in radians, is used as a nondimensional time. Figure 4 illustrates the case $\psi = 60°$, $\eta = 1.0$, $\delta = 180°$, $\mu = 0.19$, and $\Delta\psi = 30°$, where μ is the advance ratio (relative wind speed normalized by tip speed) and the tip vortex line is represented by straight line segments subtending $\Delta\psi = 30°$.

Point P_ℓ is the point at the upper (larger azimuth angle) end of the vortex-line segment whose contribution to the induced velocity at point P_η is being computed. The element of wake at point P_ℓ was originally shed from radial station ℓ on the ℓ blade, which had an azimuth angle ϕ at the instant of shedding. The current age, δ_ℓ, of the element of wake at point P_ℓ is the difference between ϕ and the current azimuth angle of the ℓ blade. On a multi-bladed rotor the ℓ blade and the η blade are not always the same. The ℓ blade is identified by ζ, which is the azimuth angle by which the ℓ blade leads the η blade. Thus, for a three-bladed rotor, ζ can have three values: 0°, 120°, and 240° associated with the three blades. For example, in Fig. 4, $\zeta = 0$ and in Fig. 5, $\zeta = 120°$.

Figure 5 illustrates the case of a three-bladed rotor $(n_b = 3)$, $\mu = .19$, $\eta = .50$, $\ell = 1.0$, $\psi = 120°$, $\delta = 0°$, $\phi = 150°$,

$\Delta\phi = 30°$, $\delta_\ell = 90°$, $\zeta = 120°$, and $m = 2$, where $\Delta\phi$ is the vortex line segment size and m is the number of turns (360° in azimuth) of wake included. Only the tip vortex of the $\zeta = 120°$ blade is shown for clarity. The ℓ blade is shown at three different ages: 0, δ_ℓ, and $\delta_\ell + \Delta\phi$ while the η blade is only shown at age $\delta = 0$.

Notice that the age δ_ℓ is related to ζ, ψ, and ϕ:

$$\delta_\ell = \psi + \zeta - \phi \tag{1.1}$$

For wake geometry computations δ is not zero at points P_η in the wake and Eq. 1.1 can be generalized:

$$\delta_\ell = \psi + \zeta + \delta - \phi \tag{1.2}$$

For airloads computations, however, δ is always zero, since the induced velocity is only computed at points P_η on the rotor blades. Since δ is not needed for the age of P_η in airloads computations, it is normally used to replace δ_ℓ, the age of P_ℓ, in these computations. Thus the δ used in wake geometry computations is always zero in airloads computations and the δ_ℓ used in wake geometry computations becomes δ in airloads computations. This means that Eq. 1.2 applies only to wake geometry computations and that, for airloads computations, Eq. 1.1 becomes:

$$\delta = \psi + \zeta - \phi \tag{1.3}$$

The points at the ends of the vortex-line segment, whose contribution to the induced velocity at point P_η is being computed, are labeled points P_a and P_b. Normally, point P_a is also point P_ℓ. The vectors from point P_η to points P_a and P_b are $\vec{a}$ and $\vec{b}$, respectively. The nondimensional circulation of the vortex-line segment is γ_a, with positive γ_a going from point P_a to point P_b. Figure 5 illustrates these definitions.

The airload at point P_η on the η blade is $L(\eta,\psi)$, with dimensions of lb/in. The local blade section angle of attack at point P_η on the η blade is $\alpha(\eta,\psi)$ radians. The total inflow, perpendicular to the TPP, at point P_η on the η blade is $\lambda(\eta,\psi)$, normalized by tip speed (ΩR). The distortion vector of a point P_η on the tip vorte. is $\vec{D}(\psi,\delta)$, normalized by rotor radius (R). Note that if point P_η is not constrained to be on the tip vortex, then $\vec{D}$ also becomes a function of the radial station η.

Since only steady-state flight is considered here, airloads, inflow, blade motion, etc., repeat every 2π in azimuth. It is, therefore, often convenient to harmonically analyze these quantities. For most variables, the harmonics are written conventionally. For example, airloads:

$$L(\eta,\psi) = L_0(\eta) + \sum_{n=1}^{\infty} \left[L_{nc}(\eta)\cos(n\psi) + L_{ns}(\eta)\sin(n\psi)\right] \tag{1.4}$$

and inflow, positive downwards:

$$\lambda(\eta,\psi) = \lambda_0(\eta) + \sum_{n=1}^{\infty} \left[\lambda_{nc}(\eta)\cos(n\psi) + \lambda_{ns}(\eta)\sin(n\psi)\right] \tag{1.5}$$

For the rotor-blade flapping angle $\beta(\psi)$, positive upwards, the NACA notation is used:

$$\beta(\psi) = a_0 - \sum_{n=1}^{\infty} \left[a_n\cos(n\psi) + b_n\sin(n\psi)\right] \tag{1.6}$$

The flapping is measured relative to the no-feathering plane (NFP), since there is no first harmonic flapping relative to the TPP.

1.5.3 Normalization

Most of the variables used in this report are nondimensional (normalized). Table 1 summarizes the general rules used in normalizing variables. Exceptions to these rules will be explained as they occur in the rest of the report. Standard helicopter normalization is used for advance ratio μ, thrust coefficient C_T, horizontal force coefficient C_H, power coefficient C_P, torque coefficient C_Q, and solidity σ.

SECTION 2

VORTEX-WAKE MODELS

Any scheme to compute either wake induced airloads or wake geometry must start with a mathematical model of the wake. As discussed previously, this model must be as simple as possible to allow efficient computation of induced velocities. The Biot-Savart relation (for large distances) is such that the magnitude of the induced velocity varies inversely with the square of the distance between a finite element of vortex wake and the point at which the induced velocity is computed. Therefore, when computing the induced velocity at a point P_η, the wake near point P_η must be treated very carefully. The wake moderately far from point P_η can be represented by relatively simple models and the wake very far from point P_η can be neglected. The Biot-Savart relation also implies a singularity in induced velocity at the center of any vortex wake element having finite circulation. Since there are no infinite induced velocities in reality, the mathematical model of the vortex wake must include physically realistic ways of avoiding these singularities.

2.1 Basic Wake Model

First, the basic wake model must be chosen. This consists of a concentrated tip vortex line, an inboard trailing vortex line or sheet, and a shed wake. The trailing wake is the wake resulting from radial variations in the bound circulation, while the shed wake comes from azimuth bound circulation variations. As discussed in Subsection 1.1, the tip vortex is the most important part of the wake and must be represented in any wake model. The inboard trailing wake is necessary to conserve circulation. Trial runs show that airloads and wake geometry computations made both with and without it included, give substantially different results (Fig. 6). If the trailing wake circulation is constant around the azimuth, then there is no shed wake. If the trailing-wake circulation varies around the azimuth, then there must be a shed wake to conserve circulation.

To extend the basic wake model all the way to infinity would require an infinite amount of computation, which is not practical. The following approximations are, therefore, used. For hovering cases, the basic wake model extends downward for about one rotor radius, and thereafter the tip vortex is represented by a semi-infinite vortex cylinder over which the Biot-Savart relation can be integrated in closed form. For forward-flight cases, enough revolutions of wake are included in the basic wake model to ensure that it contains all wake elements within about a rotor radius of any point at which induced velocities are computed. This is, typically, four revolutions for $\mu = 0.10$ or two revolutions for $\mu = 0.20$. The rest of the wake is neglected, since it makes a negligible contribution to the induced velocities being computed. Figure 7 shows two revolutions of the basic wake model.

2.2 Wake Model with n Trailing Vortex Lines

The question of more elaborate wake models than the basic one described above requires some discussion. The standard technique is to use a multiplicity of trailing vortex lines, typically 10 to 20. To determine the wake circulation, it is necessary to find the induced velocity, and hence, the circulation at n-1 radial stations for an n trailing vortex-line-wake model. When the basic wake model is used, the induced velocity can be computed at any convenient number of radial stations, typically six, which results in a factor of 1.5 to 3 saving in computation. The lesser number of wake elements in the basic wake model results in a further factor of 6 to 13 saving in computation time.

To avoid the very time-consuming, repeated computation of induced velocities by the Biot-Savart relation during airloads computations, a matrix of influence coefficients (QMAT) is stored.

When QMAT is multiplied by a bound circulation matrix, a matrix of the induced velocity at each radial and azimuthal station is produced. QMAT is only a function of wake geometry and the wake model used; thus it remains constant during the complete circulation iteration. The size of QMAT is determined by the number of induced velocities computed (number of radial stations times the number of azimuthal stations, typically 6x25 = 125) and by the size of the bound circulation matrix (the number of radial stations times the number of azimuthal stations, typically 1x25 = 25). In the basic wake model, only one radial bound circulation station is required to completely describe the wake model. Thus, at 4 bytes per word, QMAT requires about 15k bytes of core storage. When compared to the 180k bytes required by the entire airloads program, this is not excessive. For the n trailing vortex-line case, n-1 radial bound circulation stations are needed. Thus, for n=20, about 930k bytes of core storage is needed, which is more than 5 times the size of the rest of the airloads program. Thus, to save both computation time and computer core storage, effort has been concentrated on the basic wake model and simple extensions which only require one radial bound circulation station.

2.3 Wake Model Elements

It is not possible to integrate the Biot-Savart relation for induced velocity in closed form over the relatively simple "rigid wake" geometry, much less over the actual distorted wake geometry. This leaves a choice between numerical integration over the actual shape of the wake or the construction of an approximate wake model out of simple elements for which the Biot-Savart relation can be integrated in closed form. In the first case, errors arise from the computation of the integrand at discrete points in an area of rapid variation. These errors become large near a concentrated vortex line, such as the tip vortex, unless the number of integrand computation points is made very large. In the second case, errors arise from the representation of curved vortex line elements by a series of straight line segments. These errors are acceptable for reasonable line-segment lengths (typically 15° in azimuth). For a given level of accuracy, the second case requires much less computation than the first and it is, therefore, used hereafter.

The simple wake elements chosen are straight vortex-line segments, plane rectangular vortex sheet segments, and circular arc vortex-line segments. Formulas are developed in Appendix A for the velocity induced, at a general point in space, by straight vortex-line segments having either constant circulation or a linear variation of circulation. Similar formulas are developed in Appendix B for plane rectangular vortex sheets. Formulas are developed in Appendix C for the velocity induced by a circular arc vortex-line segment (with a finite vortex core) at a point on itself. The self-induced velocity of a circular arc vortex-line segment is infinite unless a vortex core, where the circulation is a function of radius, is included in the computation. The actual use of these elements to model the various parts of the wake is described below, in Subsections 2.5, 2.6, and 2.7.

2.4 Vortex Core Models

Three different vortex core models are used for vortex-line segments. One of these models is also used for vortex sheet segments. All of these models assume that the vortex core is axially symmetric about the center of the vortex line. The size of the core region is characterized by a core radius ρ_c (normalized by R). This parameter ρ_c is defined as the radius at which the maximum tangential induced velocity is achieved. Thus ρ_c is not necessarily the radius at which the vorticity goes to zero or the outer limit of the region where the Biot-Savart relation is inaccurate.

2.4.1 Solid Body Rotation Model

The simplest vortex core model is solid body rotation. For this case, the radial circulation distribution is:

$$\gamma_c(\rho) = \begin{cases} \rho^2 & , 0 \leq \rho \leq 1 \\ 1.0 & , \rho \geq 1 \end{cases} \tag{2.1}$$

where $\gamma_c(\rho)$ is the circulation at radius ρ (normalized by the total circulation of the vortex line) and ρ is the radial coordinate from the center of the vortex line (normalized by ρ_c). This model considerably overestimates the induced velocity near $\rho = 1.0$ and hence it is not used here for straight vortex-line or vortex-sheet segments. Bliss [40] shows that the choice of circulation distribution makes very little difference when computing the self-induced velocity of a vortex ring. The simple solid body rotation model is, therefore, used here for curved vortex-line segments which are simply incomplete vortex rings.

2.4.2 Fixed Wing Vortex Core Model

A somewhat more complicated model is the "fixed wing vortex core model". This model is based on measurements of tip vortex properties behind fixed-wing wind-tunnel models [36, 37, 38]. The vortex core radius is:

$$\rho_c = .05\,(c/R) \tag{2.2}$$

where c = blade chord (ft). The radial circulation distribution is:

$$\gamma_c(\rho) = \frac{\rho^2}{1+\rho^2} \tag{2.3}$$

In the experimental data [36, 37, 38], ρ_c varies from 0.025 (c/R) to 0.063 (c/R) and the peak tangentially induced velocity varies from 84% to 139% of that predicted by this model. From Eq. 2.3, $\gamma_c(1) = 0.5$ for this model and from Eq. 2.1 $\gamma_c(1) = 1.0$ for the solid body rotation model. Thus, the solid body rotation model gives twice the peak tangential induced velocity that this model does and therefore does not agree as well with the experimental data.

This fixed-wing vortex-core model is very easy to use.

For the case of a straight vortex-line segment:

$$\rho = h/\rho_c \tag{2.4}$$

where h is defined in Eq. A.19 of Appendix A. Now $\gamma_c(h)$ can be written:

$$\gamma_c(h) = \frac{h^2}{h^2 + \rho_c^2} \tag{2.5}$$

To obtain the induced velocity contribution of a straight-line vortex segment, including a vortex core, the normal expression developed in Appendix A is multiplied by $\gamma_c(h)$. The case of a vortex-sheet segment is more complicated because $\gamma_c(h)$ is a function of x and is inside the x integral. This case is discussed in Subsection B.3 of Appendix B.

2.4.3 Rotary Wing Vortex Core Model

Recently, Cook [39] has published measurements of the tip vortex of a full-scale, one-bladed helicopter rotor hovering on a whirl tower. These measurements show almost an order of magnitude smaller core radius ρ_c when compared with the fixed-wing measurements [36, 37, 38]. The magnitude of the measured peak tangentially induced velocity is about the same. This implies a very different type of radial circulation distribution and a new "rotary-wing vortex core model" has been developed to fit this data. The vortex-core radius is:

$$\rho_c = .008\,(c/R) \tag{2.6}$$

and the radial circulation distribution is:

$$\gamma_c(\rho) = \begin{cases} 0.2\,\rho^2/(1+\rho^2) & , \ 0 \le \rho \le 1 \\ 1.0 - 0.9\,e^{-.075(\rho-1)} & , \ \rho \ge 1 \end{cases} \tag{2.7}$$

This model is only used for straight vortex-line segments. The model is not used for vortex sheets because there are no measurements available for vortex sheets behind a rotor blade. In the absence of experimental data, the "fixed-wing vortex core model", which is much easier to integrate, is used for vortex-sheet segments.

2.4.4 Comparison of Vortex Core Models

A comparison of the induced velocity (V_θ) as a function of the radial coordinate (ρ) for the three different vortex-core models is given in Fig. 8. The results are normalized by the peak induced velocity of the solid body rotation model (V_θ^*). The core size ρ_c for the solid body rotation and the "fixed-wing" models is the same and is used to normalize the ρ in Fig. 8. The ρ_c for the "rotary-wing" model is only 15% as large as the ρ_c for the other two models. The smaller core size is a necessary part of the model.

For the solid body rotation model, the influence of the vortex core ends at $\rho = 1$ and everything after that is simply the Biot-Savart relation. For the other two models, the influence of the vortex core extends to infinity. As can be seen in Fig. 8, he results of both of these models are asymptotic to the Biot-Savart relation for large ρ. Therefore, the computation of the vortex core influence can be dropped for sufficiently large ρ.

As shown in Fig. 8, the fixed-wing model gives about the same results as the rotary-wing model, except for $\rho < 0.5$ (i.e., $\rho < 0.025$ chords). Since such close blade-vortex interactions do not occur very frequently, airloads computations using the two different models normally give similar results (see Fig. 9). As shown in Appendix D, there is some doubt about the validity of the rotary-wing model. Based on these considerations, and the great simplicity of the fixed-wing model, the rotary-wing model remains experimental while the fixed-wing model is standard. For more background on the fixed- and rotary-wing vortex-core models, see Appendix D.

2.5 Tip Vortex Models

The concentrated tip vortex line is (with one exception) represented by straight vortex-line segments subtending equal azimuth intervals ($\Delta\phi$, typically 15°). The exception occurs when it is necessary to compute the induced velocity at a point on the tip vortex line, due to the adjacent tip vortex-line segments. This occurs only in the tip vortex geometry program. For this case, curved vortex-line segments are used, Appendix C. Whether curved or straight, vortex-line segments have vortex cores. For curved segments, the solid body rotation vortex core model is used. For straight segments, the fixed-wing vortex-core model is normally used, but the rotary-wing vortex-core model is available as an option.

2.5.1 Vortex Core Bursting

A close interaction between a rotor blade and a concentrated tip vortex line may cause changes in the properties of the tip vortex line. One possibility is that a close blade-vortex interaction will cause instability in the viscous vortex core leading to a breakdown or bursting of the core. This bursting does not destroy circulation, but does result in a large increase in vortex-core radius.

Another possibility is the interaction of the concentrated tip vortex line and the near wake. A close blade-vortex interaction generates rapid spanwise variations of bound circulation on the rotor blade (Fig. 10). These variations lead to a trailing wake near the blade known as the near wake. The central portion of the near wake comes from the very rapid change of bound circulation between the positive and the negative peaks shown in Fig. 10. This central portion is relatively concentrated, has circulation opposite in sign to the tip vortex line, and can be of the same order of magnitude as the tip vortex line. The outer portions of the near wake are spreadout vortex sheets having the same total magnitude of circulation as the central portion, but of opposite sign. Thus, the central portion of the near wake could roll up with the tip vortex line and partially cancel it, leaving the distributed outer portions instead. This process should be modeled by reducing the tip vortex-line circulation and adding vortex sheets on either side of the vortex line. This is complicated. A crude, but much simpler model, is to increase the vortex-core radius.

To explore the effect of these possibilities on airloads and wake geometry, a vortex-bursting model has been developed. Two vortex-core radii are used for the concentrated tip vortex line: a small radius associated with the vortex-core model and a much larger, burst radius. All tip vortex-line segments start out with the small radius when they leave the blade at zero age. If the tip vortex-line segment at azimuth angle ϕ experiences, for the first time, a close blade-vortex interaction at age δ_b, it is said to have burst. This segment at ϕ will then have the small radius for all ages less than δ_b and the larger, burst radius for all ages greater than δ_b. This $\delta_b(\phi)$ function is obtained routinely during airloads computations and can be input for wake geometry computations. The criteria for determining if a blade-vortex interaction is close enough to cause bursting and at what point in the interaction this bursting occurs are input parameters. These parameters and the vortex-core radii can be varied to test different theories.

Observation of measured rotor airloads [49,50] shows that close blade vortex interactions near $\phi = 90°$ normally produce larger airload peaks than similar interactions at larger azimuth angles. Ordinary vortex bursting is not the answer, because these peaks often come from the first close interaction that a particular tip vortex-line segment experiences. These first interactions are due to a rotor blade sweeping over the tip vortex line left by the preceding blade, starting at about $\phi = 90°$ and continuing to about $\phi = 270°$. Before $\phi \simeq 90°$ and after $\phi \simeq 270°$, the tip of the blade does not extend as far as the tip vortex left by the preceding blade and no close interaction occurs. To clarify this, look at Fig. 5 and imagine another η blade at $\psi = 60°$ and $\zeta = 180°$. This blade does not reach the tip vortex line.

A possible explanation for the observed reduction in airload peaks is the propagation of vortex bursting. When the rotor blade first starts sweeping over the tip vortex from the preceding blade, at about $\phi = 90°$, normal bursting occurs. This means that the bursting occurs behind the blade, after the first close interaction. Thus, this first close interaction results in a large airload peak at about $\phi = 90°$. This bursting instability then propagates up the tip vortex for increasing ϕ at a rate faster than the rotor blade. Thus, the tip vortex can burst before the first close interaction for $\phi > 90°$. This could explain the reduced airload peaks in this area.

This propagation of bursting is speculative. On the other hand, it is the only explanation which has been developed for the observed reduction in the airload peaks, due to the first close blade-vortex interaction, over the forward portion of the rotor disk. To investigate the propagation idea, a simple model has been developed. Propagation starts when the first close interaction occurs (normally near $\phi = 90°$). A uniform propagation rate and the ϕ at which all propagation stops are input parameters which can be varied to test different theories.

To summarize, models have been developed for tip vortex-core bursting and the propagation of this bursting. The parameters defining these models are inputs which can be varied. Bursting is simulated by a simple increase in vortex-core radius, the vortex-core model remains unchanged. A segment of tip vortex which has already burst cannot be burst a second time by any subsequent close blade-vortex interactions.

2.5.2 Near Wake Caused by the Tip Vortex

Close blade-vortex interactions, which involve the concentrated tip vortex line, lead to extremely rapid spanwise variations of the bound circulation. This causes the shedding of some extra trailing wake which, in turn, modifies the bound circulation distribution by changing the induced angle-of-attack distribution. This extra trailing wake ultimately either rolls up with the tip vortex or becomes part of the distributed inboard trailing wake. The part of this wake just behind the blade which shed it (the near wake) has not had time to roll up and/or merge with the rest of the wake. This near wake, therefore, has an important effect on the airloads distribution, and hence the bound circulation distribution,which is not represented by the basic wake model. In addition to this near wake effect, Johnson [52] has shown that there are important lifting-surface effects on the airloads due to a close blade-vortex interaction.

The normal solution to this near wake problem is to use lifting-line theory with an n trailing vortex-line model. This solution is expensive and it neglects both lifting-surface effects and any rollup of the near wake. Furthermore, the n used is often too small to adequately represent the near wake due to a close blade-vortex interaction. For example, consider Fig. 10 which shows a distance of less than one chord between the positive and negative bound circulation peaks. Since the peaks are located randomly with respect to the n trailing vortex lines, these lines must be spaced no more than 1/3 chord apart to ensure hitting the peaks. This implies values of n around 50 for typical high aspect ratio rotor blades, which is very expensive.

A common solution to the rollup problem is to only use the n trailing vortex line model for the near wake, with a two trailing vortex line model (similar to the basic wake model) for the far wake. This is good as far as it goes, but it still requires the computation of induced velocities at (n-1) radial stations and neglects lifting-surface effects. Lifting-surface theory could be applied to this model, but it would require the computation of induced velocities at several chordwise stations for each radial station. Taking a minimal 5 chordwise stations and (n-1) = 50, this implies the computation of induced velocities at 250 stations on the blade at each azimuth. This compares with the 6 stations normally used in this report.

To avoid this expense, superposition is used and the near wake due to each close blade-vortex interaction is treated separately. Johnson [52,53] has developed a planar lifting-surface solution for the loading on an infinite aspect ratio wing, due to an infinite, straight vortex line in a subsonic, compressible, free stream. The vortex line lies in a plane parallel to the plane of the wing, at a given distance below it and at an arbitrary angle with the wing centerline. Based on this solution, Johnson [52, 53] has developed an approximate analytical expression for the lift influence function. A simplified, incompressible version of this approximate expression is used for each individual close blade-vortex interaction. Since the tip vortex line is modelled by a series of straight-line segments, each of these segments is examined individually to see if it is involved in a close blade-vortex interaction.

The rotor blade has a finite aspect ratio, the vortex-line segment has a finite length, and the plane of the vortex line is not always parallel to the plane of the rotor blade, but all of these factors are neglected in applying this approximate lifting-surface solution. Furthermore, the lift-influence function is applied to the induced velocity contribution of the vortex-line segment instead of the lift. This implies that the lift-curve slope is constant. Notwithstanding all of these approximations this simplified lifting-surface solution is much more representative of near wake effects than a lifting-line solution. Furthermore, by applying this lifting-surface solution only at points where it becomes important, a very efficient computation is achieved.

In summary, each tip vortex-line segment is examined to see if it is involved in a close blade-vortex interaction. If it is, then its contribution to the induced velocity at various points on the blade is multiplied by an approximate lift influence function developed by Johnson [52, 53]. Superposition is assumed to hold and the near wake due to anything but close blade-vortex interactions is neglected, except for a very approximate representation by the basic wake model and its extensions.

2.5.3 Tip Vortex Sheet Plus Vortex Line Model

In reality, the trailing wake is shed from the blade as a concentrated vortex line plus some distributed circulation. The concentrated vortex line is shed from the tip of the blade. The distributed circulation is shed inboard of the tip. Some of the distributed circulation eventually rolls up with the concentrated tip vortex. The rest of the distributed circulation remains spread out, at least near the rotor. In this work, the tip vortex is defined to include the concentrated tip vortex line, plus any distributed circulation which eventually rolls up with it. The rest of the distributed circulation is the inboard trailing wake.

In the basic wake model, it is assumed that, at each azimuth, all of the trailing wake shed outboard of the radial maximum in bound circulation will eventually roll up with the tip vortex (see Appendix D.4 for theoretical justification of this assumption). The normal basic wake model assumes that this rollup is instantaneous and that the concentrated tip vortex appears, full strength, at the tip of the blade. To test this assumption of instantaneous rollup, an alternative vortex sheet plus vortex line (VSVL) tip vortex model is used. This VSVL model represents both the concentrated and the distributed parts of the tip vortex, as well as the rollup process. The representation used is very simple, and not really suitable for a production program, but it is adequate to evaluate the importance of rollup in airloads computations. The airloads program can use either the simple basic wake model (instantaneous rollup) or the VSVL tip vortex model. The wake geometry program uses only the simple instantaneous rollup model.

The tip vortex circulation, in the VSVL model, is initially split between a concentrated tip vortex line and a distributed tip vortex sheet. The concentrated tip vortex line is shed from the blade tip ($\ell = 1.0$). The distributed vortex sheet is shed from a segment of the blade extending from the tip ($\ell = 1.0$) inboard to a radial station $\ell_t(\phi)$. This $\ell_t(\phi)$ is an input function of azimuth angle (ϕ) and is determined using the bound circulation distribution from a previous airloads computation for the same case.

The total circulation of the tip vortex line plus the tip vortex sheet is the same as the circulation of the tip vortex line in the simple instantaneous rollup model. The fraction of the total tip vortex circulation which is included in the concentrated tip vortex line at azimuth angle ϕ and age δ is $f_t(\phi,\delta)$. For the instantaneous rollup model $f_t(\phi,\delta) = 1.0$ for all ϕ and δ.

For the VSVL Model:

$$f_t(\phi,\delta) = \text{Min.}\begin{cases} 1.0 \\ f_t^{o}(\phi) + \delta\, f_t^{D} \end{cases} \tag{2.8}$$

Thus, $f_t(\phi,\delta)$ is assumed to be linear in age (δ) starting from an initial $f_t^{o}(\phi) = f_t(\phi,0)$ at the instant the wake element leaves the rotor blade ($\delta = 0$). The function $f_t^{o}(\phi)$ is an input to the program and, like $\ell_t(\phi)$, is determined using the bound circulation distribution from a previous airloads computation for the same case. The rollup rate (f_t^{D}) is also an input to the program and is varied as a parameter.

The near wake segment of the tip vortex sheet is the segment just behind the η blade. This near wake must be handled with special care, due to its proximity to the points P_η on the η blade at which the induced velcoity is being computed. To avoid the edge singularity, the radial station $\ell_t(\phi)$ of the inner edge of the vortex sheet must always be chosen midway between adjacent points P_η. In addition, the geometry of the near tip vortex sheet must be very accurately defined relative to the η blade.

Normally, the geometry of a trailing vortex sheet segment is determined by translating a trailing vortex-line segment a distance Δx perpendicular to itself (see Appendix B). Since trailing vortex-line segments are not always perpendicular to the ℓ blade which shed them, this can lead to either a gap or an overlap between the ℓ blade and the vortex sheet (see Fig. 11). Ordinarily, this is not important, but for the near wake, the ℓ blade is also the η blade, on which the induced velocity is being computed. So for the near wake case, no gap or overlap between the η blade and the vortex sheet can be allowed. The near wake vortex-sheet segment, therefore, extends back perpendicular to the η blade for a distance equal to the corresponding trailing vortex line (see Fig. 11).

The net result of the VSVL model is that the tip vortex sheet circulation decays linearly with time, while total tip vortex circulation remains constant. The dimensions of the tip vortex sheet do not change with time. A model where the vortex sheet rolls up into the vortex line like a window shade would be closer to what happens in reality. The VSVL model is much simpler to implement, however, and it is good enough to test the importance of modeling tip vortex rollup at all.

2.6 Inboard Trailing Wake

The inboard trailing wake is the wake caused by radial variations in bound circulation which is shed from that portion of the rotor blade inboard of the peak bound circulation. It has a total circulation equal in magnitude but opposite in sign to that of the tip vortex. Unlike the tip vortex, the inboard trailing wake does not seem to roll up into a concentrated vortex line (at least not near enough to the rotor to affect the rotor airloads). This distributed nature of the inboard trailing wake means that its precise geometry is not as important as the geometry of the concentrated tip vortex line. This is fortunate because the computation of its geometry would involve the computation of the location of a network of points over a wide area. Compared to the computation of the location of points on a tip vortex line, this would be very expensive. For these reasons,only the simple "rigid wake" geometry is used for the inboard trailing wake in this report, with the exception of an empirical "prescribed" wake geometry used for some hovering cases.

Ideally, the inboard trailing wake should be represented by a vortex sheet of appropriately varying circulation. In practice, this vortex sheet is broken up into either vortex-line or vortex-sheet segments as described in Subsection 2.3. Azimuthally, these segments either have constant circulation or linearly varying circulation. Radial circulation variations could be represented by a multitude of vortex-line or vortex-sheet segments. As discussed in Subsection 2.2, this alternative is very expensive and is not used here. Instead, either a single vortex line or a single vortex sheet is used.

When the vortex-sheet model is used, the circulation of the vortex sheet does not vary with radius. This corresponds to a linear variation in bound circulation with radius. The inboard edge of the vortex sheet is normally taken to be at the blade root cutout radius. The outboard edge of the vortex sheet is an input function of azimuth angle $\ell_i(\phi)$ for airloads computations, while it is computed in wake geometry computations. In either case, it is a function of the bound circulation distribution obtained from a previous airloads computation, if available. This vortex-sheet model works well, but it is more expensive than the vortex-line model, due to the more complicated formula for the induced velocity contribution of a vortex-sheet segment.

The single vortex-line model is faster and simpler to use. However, it has the disadvantage of using a concentrated vortex line to represent a distributed vortex sheet. This can lead to close blade-vortex interactions which cause large airloads peaks. These peaks are products of the model and do not exist in reality where there is no concentrated vortex line to cause them. This problem can be alleviated by using a vortex line having a very large vortex-core radius. Normally, this vortex line is located radially in the middle of the average vortex sheet and has a vortex-core diameter equal to the width of this average vortex sheet. Basically, this corresponds to spreading out the vortex-line circulation over a circle whose diameter is equal to the width of the vortex sheet. Naturally, this three-dimensional spreading out of the circulation does not give the same answer as the two-dimensional spreading involved in a vortex sheet. However, the approximation is adequate for most purposes and this model is considerably less expensive than the vortex-sheet model. Since the vortex-core radius is artificial, there is no vortex core bursting.

The near portion of the inboard trailing wake (that part just behind the η blade) must be handled with some care. For the vortex-line model, it is necessary only to make sure that the center of the vortex line starts at a radial station midway between two adjacent points P_η at which airloads are computed. For the vortex-sheet model, the edges of the sheet must start at radial stations midway between two adjacent points P_η. In addition, the geometry of the vortex sheet must be very accurately defined relative to the η blade. This is covered in more detail for the near tip vortex sheet case in Subsection 2.5.3.

2.7 Shed Wake

The shed wake is all of the wake caused by azimuthal variations in the bound circulation. Like the inboard trailing wake, it is a distributed vortex sheet which does not appear to roll up near the rotor. Furthermore, the magnitude of the circulation in the shed wake is much less, sometimes as much as an order of magnitude less, than the trailing-wake circulation. Thus, the shed wake, with the exception of the near shed wake, can be treated fairly simply. The near shed wake must be handled carefully, due to the singularity as a shed vortex line approaches the points P_η on the η blade.

Normally, a vortex-line model is used for the shed wake with the exception of the near shed wake. This model consists of accumulating all of the change in bound circulation between azimuth angles ϕ and $\phi + \Delta\phi$ into a shed vortex line at azimuth angle ϕ. The shed vortex line extends from a corner between two tip vortex-line segments to a corner between two inboard trailing vortex-line segments (see Fig. 7). To simulate the distributed nature of the shed wake an artificially large vortex-core radius, equal to half the distance between adjacent shed vortex lines, is used. A slightly more accurate model would place the shed vortex line at $\phi + \Delta\phi/2$ instead of at ϕ. However, this requires the use of vortex-line segment end points which are different from those used for the trailing wake. The increased accuracy is not worth the extra complication.

When a more accurate model is desired, the shed vortex-sheet model is used. This vortex sheet is generated by translating the shed vortex line forward, perpendicular to itself, from ϕ to $\phi + \Delta\phi$. This model is considerably more expensive than the shed vortex-line model and is used only for exceptional cases.

The near shed wake is always modeled by a vortex sheet, due to its proximity to the points P_η on the η blade. If this near shed vortex-sheet segment were extended all of the way up to the points P_η, there would be a singularity in the induced velocity at points P_η. Miller [9] presents a plot of ε_s vs. reduced frequency ($k = \frac{\omega b}{V}$), where ε_s is the limit of the shed-wake integration for use in the lifting-line approximation. This plot was developed using classical two-dimensional unsteady aerodynamics. For typical helicopter reduced frequencies ε_s is about 0.25 chord. Thus, the near shed wake vortex-sheet segment is generated by constructing a shed vortex-line segment ε_s behind the points P_η on the η blade. This vortex-line segment is then translated backwards, perpendicular to itself, from $\psi - \varepsilon_s$ to $\psi - \Delta\psi$ to generate the near vortex-sheet segment.

2.8 Determining the Circulation

The various vortex wake elements which make up the wake models have been described. Now the circulation of these wake elements must be determined. The circulation of the wake elements is determined, by conservation of circulation, from a bound circulation model. Circulation is conserved with two exceptions:

1) Wake elements which are very far away from point P_η are not included in the model and their contribution to the induced velocity at point P_η is neglected.

2) As an experiment, some cases were run where the inboard trailing wake was neglected: this approach was not a success.

The bound circulation model is normally based on the bound circulation distribution $\gamma_n(\eta,\phi)$ obtained from a previous iteration of the airloads computation. For the zeroth iteration, no such bound circulation distribution is available and the bound circulation is assumed to have a constant average value γ_o, based on the thrust of the rotor. This means that there is no shed wake for this case and that all trailing wake elements have the same circulation.

The bound circulation $\gamma_n(\eta,\phi)$ is computed at n_η (normally $n_\eta = 6$) radial stations and n_ψ (normally $n_\psi = 25$) azimuthal stations. For the basic wake model there is only one value of bound circulation $\gamma_{bv}(\phi)$ at each azimuth. This saves computer time and core storage as discussed in Subsection 2.2. Thus, the radial bound-circulation distribution $\gamma_n(\eta,\phi)$, which is computed at n_η radial stations, must be modeled by a much simpler distribution which involves storing the value of only one bound circulation $\gamma_{bv}(\phi)$ for each azimuth. There are different radial bound-circulation-distribution models, depending on the use of a vortex line or a vortex-sheet model of the inboard trailing wake and on the use of a tip vortex line only or a tip vortex sheet plus line model.

2.8.1 Vortex Line Trailing-Wake Models

For the simple basic wake model, where both the tip vortex and the inboard trailing wake are modeled by vortex lines, the radial bound-circulation model of Fig. 12 is used. The bound circulation $\gamma_{bv}(\phi)$ is chosen to be the maximum bound circulation $\gamma_n(\eta,\phi)$ at azimuth angle ϕ for all η greater than η_{MEP} (normally $\eta_{MEP} \simeq 0.5$). Since the radial stations η are spaced rather widely, a peak in $\gamma_n(\eta,\phi)$ sometimes falls in between two adjacent radial stations η. In this case, interpolation is used to find $\gamma_{bv}(\phi)$ based on $\gamma_{bv}(\phi + \Delta\phi)$ and $\gamma_{bv}(\phi - \Delta\phi)$. The value of ℓ_2 is chosen to be midway between two adjacent points P_η and to be about midway between the root cutout (ℓ_1) of the blade and the average $\ell_i(\phi)$ which would have been chosen if a vortex-sheet model had been used, see Subsection 2.6.

From Fig. 12, it is clear that the shed vortex-line circulation $\gamma_s(\phi)$ is constant radially with this model. The azimuthal variation of $\gamma_s(\phi)$ depends on $\gamma_{bv}(\phi)$:

$$\gamma_s(\phi) = \gamma_{bv}(\phi + \Delta\phi) - \gamma_{bv}(\phi) \tag{2.9}$$

If the shed vortex-line model is used, the trailing-wake circulation will be constant over each vortex-line segment and will vary between segments. For the tip vortex-line segment at azimith angle ϕ:

$$\gamma_a = \gamma_b = \gamma_{bv}(\phi) \tag{2.10}$$

and for the inboard trailing vortex-line segment:

$$\gamma_a = \gamma_b = -\gamma_{bv}(\phi) \tag{2.11}$$

If the shed vortex-sheet model is used, its circulation is constant both azimuthally and radially for each segment. This means that the trailing-wake circulation must vary linearly over each segment to conserve circulation. The values of γ_a remain as in Eqs. 2.10 and 2.11 and γ_b becomes:

$$\gamma_b = \gamma_{bv}(\phi - \Delta\phi) \tag{2.12}$$

for the tip vortex-line segment and

$$\gamma_b = -\gamma_{bv}(\phi - \Delta\phi) \tag{2.13}$$

for the inboard trailing vortex-line segment. The γ_a and γ_b used here are the circulations at the two ends of a vortex-line segment, see Appendix A.

2.8.2 Vortex-Sheet Trailing-Wake Models

For the modified basic wake model, where both the tip vortex and the inboard trailing wake are modeled by vortex sheets, the radial bound circulation model of Fig. 13 is used. As before, the bound circulation $\gamma_{bv}(\phi)$ is the maximum $\gamma_n(\eta,\phi)$ and interpolation is used as necessary. The values of $\ell_i(\phi)$ and $\ell_t(\phi)$ are chosen, outside of the computer, based on the $\gamma_n(\eta,\phi)$ from a previous iteration. The value of ℓ is normally the root cutout of the rotor blade.

From Fig. 13, it is clear that the shed-wake circulation $\gamma_s(\phi)$ should be constant between $\ell_i(\phi)$ and $\ell_t(\phi)$ and should vary linearly between ℓ_1 and $\ell_i(\phi)$ and between $\ell_t(\phi)$ and the tip ($\ell = 1.0$). This would require three shed-wake vortex-line segments at each azimuth angle ϕ, which is expensive. The far-shed wake is not important enough to justify this expense. The normal solution is to use a single shed vortex-line (or sheet) segment which runs from ℓ_2 to the tip ($\ell = 1.0$) and which has a constant circulation radially, given by Eq. 2.9. For some wake geometry computations, this constant circulation shed vortex line runs only from $\ell_i(\phi)$ to the tip ($\ell = 1.0$). There is then another shed vortex line running form ℓ_1 to $\ell_i(\phi)$ which has a circulation varying linearly from 0.0 to $\gamma_s(\phi)$. This two-segment model is more expensive than the one segment model and is not used very often. The value of ℓ_2, used in the one-segment model is normally midway between ℓ_1 and the average $\ell_i(\phi)$.

If the shed vortex-line model is used, the trailing-wake circulation will be constant over each trailing-wake segment and will vary between segments. For the tip vortex-line segment at azimuth angle ϕ:

$$\gamma_a = \gamma_b = f_t(\phi,\delta) * \gamma_{bv}(\phi) \tag{2.14}$$

The total tip vortex-sheet-segment circulation will be:

$$\gamma_a = \gamma_b = [1.0 - f_t(\phi, \delta)] * \gamma_{bv}(\phi) \tag{2.15}$$

The total circulation of the inboard trailing-vortex-sheet segment is given by Eq. 2.11. Both of the vortex-sheet segments are of constant circulation radially and azimuthally.

If a shed vortex-sheet model is used, then the trailing-wake circulation must vary linearly over each segment to conserve circulation. This model is not normally used, but if necessary, the equations for γ_b can readily be derived from observation of the examples above.

It is possible to use the inboard trailing-vortex sheet with a tip vortex-line model or to use a tip vortex sheet plus vortex line model with an inboard trailing vortex line. For these cases, the radial bound circulation models of Figs. 12 and 13 can be mixed as necessary. The circulation can be computed using appropriate selections from Eqs. 2.10 through 2.15.

It should be observed that all of the circulation models given in this section depend on only one $\gamma_{bv}(\phi)$ at each azimuth. This saves computer time and core storage as discussed in Subsection 2.2. It might be desirable to change the $\ell_i(\phi)$ and $\ell_t(\phi)$ in the computer during the circulation iteration, since these variables are functions of $\gamma_n(\eta,\phi)$. This amounts to changing the wake geometry and as discussed in Subsection 2.2, this would make circulation iterations much more expensive. On these grounds, it is suggested that $\ell_i(\phi)$ and $\ell_t(\phi)$ should be allowed to change much less often than every circulation iteration. In the current program, they are only changed externally between computer runs.

SECTION 3

ROTOR HARMONIC AIRLOADS COMPUTATION

A rotor harmonic airloads computation scheme, and a computer program (LDS-73) which uses that scheme have been developed. The airloads computation has two functions. It provides the angle of attack $\alpha(\eta,\psi)$ which the wake geometry computation (WG-71) uses to compute bound circulation $\gamma_n(\eta,\psi)$, and it is used to evaluate the influence of tip-vortex distortion on rotor harmonic-airloads computations. The airloads computation is not concerned with rotor blade structural response to the harmonic airloads, except where the response is large enough to change the airloads. The airloads computation is primarily intended for use at low advance ratios where the influence of tip-vortex distortion on harmonic airloads is expected to be most important. For these reasons, the blade-dynamics modeling and dynamic-stall modeling is relatively simple, while the aerodynamic modeling is sophisticated. It is anticipated that an industrial user of this work would use the aerodynamic models, but would provide his own blade-dynamics and blade-section aerodynamics to whatever level of sophistication is necessary.

The airloads program LDS-73 computes the rotor airloads for a given wake geometry. If the effects of tip-vortex distortion are to be included in the computation, then LDS-73 is used together with WG-71. This program computes the tip-vortex distortion $\vec{D}(\psi,\delta)$ for a given bound circulation distribution $\gamma_n(\eta,\psi)$. The relationship between LDS-73 and WG-71 is shown in Fig. 3.

The basic inputs to LDS-73 are a physical description of the rotor, the operating condition of the rotor, the initial bound circulation distribution (which can be simply a constant, γ_o), and a group of parameters indicating which wake model and which of the many different computation options are to be used. In addition, the tip-vortex distortion $\vec{D}(\psi,\delta)$ and a set of experimental airloads

XLDS(η,ψ) can be introduced as input, if necessary. The outputs are the airloads distribution L(η,ψ), various overall performance parameters (such as rotor L/D_E), rotor blade motion, the rotor downwash distribution $\lambda(\eta,\psi)$, the rotor angle-of-attack distribution $\alpha(\eta,\psi)$, the bound circulation distribution $\gamma_n(\eta,\psi)$, and a mass of detailed data which supports these results. For a detailed account of these inputs and outputs, see Appendix E.

During the evaluation of the effects of tip-vortex distortion on rotor harmonic airloads, the vital importance of close blade-vortex interactions became evident. A substantial program of fundamental research will be required to fully understand this problem. To help guide such research, an evaluation has been made here of the effect on rotor harmonic airloads of various close blade-vortex interaction phenomena. These phenomena include the tip-vortex core, tip vortex-core bursting, the propagation of tip-vortex-core bursting, lifting-surface effects, near wake effects, and local stall due to large radial pressure gradients. Most of these phenomena are modeled very simply, which is appropriate for our current level of understanding. Further investigation may show that some of these phenomena do not exist in the form postulated here, or it may show that other phenomena, not considered here, are important. It is clear, however, that some sort of close blade-vortex interaction phenomena exist which have an important effect on rotor-blade harmonic airloads, especially when tip-vortex distortion is included.

3.1 Bound-Circulation Computation

3.1.1 Bound-Circulation Iteration

A simplified flow chart for LDS-73 is given in Fig. 14. This flow chart emphasizes the basic bound circulation iteration. The program starts by reading input data and initializing various constants. Then the bound-circulation distribution for the nth iteration $\gamma_n(\eta,\psi)$ is computed. There are two basic options here, depending on whether or not an angle-of-attack distribution $\alpha(\eta,\psi)$ is available from a previous run of LDS-73. If it is not available, then this is a zeroth iteration and $\gamma_o(\eta,\psi)$ is equal to a constant γ_o, which is an input. If $\alpha(\eta,\psi)$ is available, then this is a first iteration and $\gamma_1(\eta,\psi)$ is computed from $\alpha(\eta,\psi)$:

$$\gamma_1(\eta,\psi) = \frac{a}{2\pi}\left(\eta + \mu \sin\psi\right) \alpha(\eta,\psi) \tag{3.1}$$

where $a = \frac{\partial C_L}{\partial \alpha}$ is the lift-curve slope, and $\mu = \frac{V}{\Omega R}$ is the advance ratio.

The basic wake model bound circulation $\gamma_{bv}(\phi)$ is computed from $\gamma_n(\eta,\psi)$. For a zeroth iteration, $\gamma_{bv}(\phi)$ is simply equal to γ_o. Otherwise, $\gamma_{bv}(\phi)$ is the radial peak $\gamma_n(\eta,\psi)$, with interpolation used as necessary, see Subsection 2.8. This $\gamma_{bv}(\phi)$ is used to compute the downwash distribution $\lambda(\eta,\phi)$. Subroutine LOADS uses $\lambda(\eta,\psi)$ to compute blade motion and airloads. One result of this airloads computation is a new angle-of-attack distribution $\alpha(\eta,\psi)$. This ends an iteration.

To start a new iteration, $\alpha(\eta,\psi)$ is used to compute $\gamma_{n+1}(\eta,\psi)$. If this is the first iteration, i.e., there is no old $\gamma_n(\eta,\psi)$, except $\gamma_o(\eta,\psi) = \gamma_o$, then Eq. 3.1 is used. Otherwise, a weighted average of the $\gamma(\eta,\psi)$ obtained from the new $\alpha(\eta,\psi)$ and the old $\gamma_n(\eta,\psi)$ is used:

$$\gamma_{n+1}(\eta,\psi) = w_f\left(\frac{a}{2\pi}\right)\left(\eta + \mu \sin\psi\right) \alpha(\eta,\psi) + (1 - w_f)\, \gamma_n(\eta,\psi) \tag{3.2}$$

where w_f is an input; normally $w_f = 0.5$. The iteration then proceeds as described above. After the last iteration is completed and a final $\alpha(\eta,\psi)$ is obtained, there is a final $\gamma_{n+1}(\eta,\psi)$ computation using Eq. 3.2. This $\gamma_{n+1}(\eta,\psi)$ and the corresponding $\gamma_{bv}(\phi)$ is printed and plotted by LDS-73 for comparison with $\gamma_n(\eta,\psi)$ to determine convergence.

There is no test for bound circulation convergence in the program LDS-73. The number of iterations to be done is an input.

Normally, five iterations are sufficient. If QMAT is not recomputed, and if most of the output for the intermediate iterations is suppressed (see Appendix E), then extra iterations are not expensive. The normal test for convergence is the output comparison plot of the last and next to the last $\gamma_{bv}(\phi)/\gamma_o$ (see Appendix E).

3.1.2 Bound-Circulation Models

The bound circulation models are covered in Subsection 2.8. The far-shed vortex sheet model of Subseciton 2.8 is not used in LDS-73. This is because runs were made using this model which showed no advantage over the simpler shed-vortex-line model.

One point which may cause confusion is the use of $\gamma_n(\eta,\phi)$ in Subsection 2.8 vs. $\gamma_n(\eta,\psi)$ in Subsection 3.1. Also $\gamma_{bv}(\phi)$ is obtained from $\gamma_n(\eta,\psi)$ in Subsection 3.1. The problem is that $\gamma_n(\eta,\psi)$ is computed at points P_η, which are at azimuth angle ψ. Then $\gamma_n(\eta,\phi)$ or $\gamma_{bv}(\phi)$ are used to define the circulation of a wake element at azimuth angle ϕ. Thus the function γ_n remains the same, only the azimuth angle at which γ_n is evaluated changes. This same situation will occur in WG-71 when the distortion $\vec{D}$ is computed: $\vec{D}(\psi,\delta)$, or is used: $\vec{D}(\phi,\delta_\ell)$. In LDS-73, the distortion is read in as $\vec{D}(\psi,\delta)$ and is used as $\vec{D}(\phi,\delta)$.

3.2 Induced-Velocity Computation

The computation of the induced velocity at points P_η on the rotor is the heart of the airloads computation. The induced contribution to the downwash $\lambda(\eta,\psi)$ is the source of high harmonic airloads. The ability to predict these high harmonic airloads is the chief object of this work. Also, the induced velocity computation is the most time-consuming part of the airloads computation. For a typical four-bladed rotor case run on LDS-73, the induced velocity computation takes about 30 seconds compared to less than one second for the rest of a circulation iteration. Subsequent iterations take less than one second each, unless QMAT is recomputed (see Subsection 3.2.2).

The induced velocity influence coefficient matrix QMAT is a function of the wake model and the wake geometry relative to points P_η on the rotor. If QMAT is to be held constant during a complete airloads computation, then these items must also be constant. The distortion $\vec{D}$ is constant during an airloads computation by LDS-73, because $\vec{D}$ is only computed by WG-71, a separate program. However, blade motion could still change the location of point P_η relative to the wake. The effect of zeroth harmonic flapping a_o is included by input of a constant value of a_o based on the thrust coefficient C_T:

$$a_o = \frac{3}{4}\left(\frac{LN}{2\pi}\right)\left(\frac{C_T}{\sigma}\right) \tag{3.3}$$

where σ = rotor solidity and LN = Lock Number = $\frac{2\pi\rho c R^4}{I_b}$. This is normally adequate. A more complicated expression with twist and downwash effects is given in Eq. 3.124. The effect of first harmonic flapping is included by computing the wake geometry relative to the tip path plane (TPP) and using a constant i, where i is the incidence of the TPP to the relative wind (Fig. 15). At low advance ratios, the thrust T is very nearly perpendicular to the TPP (C_H, the horizontal force coefficient, is computed by LDS-73 to check on this). Thus, i is easily calculated, if the rotor propulsive force X is known:

$$i = \sin^{-1}(X/T) \tag{3.4}$$

The effect of higher harmonics of flapping and of all other blade motion on wake geometry is neglected.

The requirement that the wake model remain constant has a more serious effect on the airloads computation. It means that the parameters $\ell_i(\phi)$, $\ell_t(\phi)$, $f_t^o(\phi)$, and f_t^D, which define the inboard trailing and tip vortex sheets (Subsections 2.5 and 2.6), cannot be changed during an airloads computation, unless QMAT is recomputed. In LDS-73, these parameters are inputs which remain constant during a run. However, they are a function of the bound circulation distribution $\gamma_n(\eta,\psi)$. In a program more advanced than LDS-73, it might be desirable to compute these parameters internally as part of the $\gamma_n(\eta,\psi)$ iteration. This will require either abandoning QMAT or greatly expanding it to include several different radial stations. This will require a lot of core storage and may not be worth the cost.

3.2.1 Induced Velocity Influence Coefficient Matrix QMAT

To save computation time, the induced part of $\lambda(\eta,\psi)$ is split into two pieces. One is an influence coefficient matrix QMAT, which only depends on the geometry of the wake relative to the points P_η on the rotor, and on the wake model. The other is the wake circulation distribution which only depends on the bound circulation $\gamma_{bv}(\phi)$. If the wake geometry and the wake model are not changed during a complete airloads computation, then QMAT only needs to be computed once for the initial iteration. Since the computation of QMAT is typically 30 times as long as the rest of an iteration, this is an important saving.

QMAT is a large matrix and it is important to minimize the size of QMAT to save computer core storage. This QMAT storage problem is one of the reasons for the choice of the basic wake model over the n trailing vortex-line model, Subseciton 2.2. To minimize the size of QMAT, consider its components. First, QMAT is a function of the location of point P_η, where the induced velocity is computed. This location is specified by η and ψ, see Subsection 1.5. Normally, there are six radial stations η and twenty-five azimuth stations ψ.

QMAT is also a function of the circulation distribution in the wake. However, in the basic wake model, the circulation of each wake element is completely determined by the bound circulation $\gamma_{bv}(\phi)$ for a fixed wake model. There are normally at least three wake elements per rotor blade for each $\gamma_{bv}(\phi)$: tip vortex, inboard trailing wake, and shed wake. Also $\gamma_{bv}(\phi)$ is cyclic every 2π in ϕ, due to the assumption of steady-state flight, Subsection 1.4. Thus, QMAT storage is minimized by summing QMAT over the various rotor blades (ζ), the various wake spirals (2π in azimuth), and the different wake elements (ℓ). In this context ℓ_1 refers to the shed wake, ℓ_2 refers to the inboard trailing wake, and ℓ_3 refers to the tip vortex. This leaves QMAT a function of η, ψ, and ϕ, where ϕ is cyclic over every 2π interval. Typically, there are twenty-five azimuth stations ϕ; thus, QMAT requires $6\eta*25\psi*25\phi= 3750$ words of storage (about 15k bytes).

Now QMAT(η,ψ,ϕ) can be defined such that the distribution of downwash perpendicular to the TPP is:

$$\lambda(\eta,\psi) = \mu \tan i + \left(\frac{b}{2R}\right) \sum_{\phi} \left[\gamma_{bv}(\phi) * \mathrm{QMAT}(\eta,\psi,\phi)\right] \qquad (3.5)$$

The μtani term is the component of the relative wind perpendicular to the TPP. The factor b/2R appears because QMAT is normalized by $\Gamma/4\pi R$, while $\gamma_{bv}(\phi)$ is given by $\Gamma/2\pi b\Omega R$ and $\lambda(\eta,\psi)$ is normalized by ΩR. This normalization of QMAT is chosen to avoid multiplying each individual entry in the QMAT matrix by b/2R. In LDS-73, the $\gamma_{bv}(\phi)$ are all multiplied by b/2R as they are computed. This minimizes the number of multiplications.

Define $q_k(\eta,\psi,\zeta,\ell,\phi)$ as the contribution of the wake element identified by ζ,ℓ, and ϕ (see Subsection 1.5) to the induced velocity component perpendicular to the TPP at point P_η. Let $q_k(\eta,\psi,\zeta,\ell,\phi)$ be normalized by $\Gamma/4\pi R$, where Γ is the circulation of the wake element (ft^2/sec). Now QMAT(η,ψ,ϕ) can be computed:

$$\mathrm{QMAT}(\eta,\psi,\phi) = \sum_{\zeta} \sum_{n=0}^{m-1} \Bigg[\sum_{\ell=\ell_2}^{\ell_3} q_k(\eta,\psi,\zeta,\ell,\phi+2\pi n) + q_k(\eta,\psi,\zeta,\ell_1,\phi-\Delta\phi+2\pi n) - q_k(\eta,\psi,\zeta,\ell_1,\phi+2\pi n) \Bigg] \qquad (3.6)$$

The two ℓ_1 terms arise because the shed wake element circulation is:

$$\gamma_s(\phi) = \gamma_{bv}(\phi + \Delta\phi) - \gamma_{bv}(\phi) \tag{2.9}$$

The summation over n is due to the cyclic nature of $\gamma_{bv}(\phi)$ and hence of QMAT (η,ψ,ϕ). The upper limit m is the number of spirals (revolutions) of wake included in the wake model.

3.2.2 Induced Velocity SUM Matrix

The SUM matrix is the matrix of $q_z(\eta,\psi,\zeta,\ell)$, which is called SUM in LDS-73. This $q_z(\eta,\psi,\zeta,\ell)$ is the contribution of all of the wake associated with a particular rotor blade (ζ), and type of wake (ℓ): shed wake, inboard trailing wake, or tip vortex, to $\lambda(\eta,\psi)$. It is computed by summing the contributions of the various wake elements over ϕ:

$$q_z(\eta,\psi,\zeta,\ell) = \left(\frac{b}{2R}\right)\sum_{\delta=0}^{2\pi m} \gamma_a * q_k'(\eta,\psi,\zeta,\ell,\delta) \tag{3.7}$$

where γ_a is the wake element circulation (see Subsection 2.8). The function $q_k'(\eta,\psi,\zeta,\ell,\delta)$ is the same as $q_k(\eta,\psi,\zeta,\ell,\phi)$, except that the age of point P_ℓ (δ) has replaced ϕ. Recall from Subsection 1.5

$$\delta = \psi + \zeta - \phi \tag{1..3}$$

This SUM matrix can now be used to compute $\lambda(\eta,\psi)$:

$$\lambda(\eta,\psi) = \mu \tan i + \sum_{\zeta}\sum_{\ell} q_z(\eta,\psi,\zeta,\ell) \tag{3.8}$$

This computation of $\lambda(\eta,\psi)$ is done as a check on the results of Eq. 3.5 whenever $q_z(\eta,\psi,\zeta,\ell)$ is available. Since QMAT (η,ψ,ϕ) has already been summed over ζ and ℓ, $q_z(\eta,\psi,\zeta,\ell)$ cannot be obtained from QMAT(η,ψ,ϕ). Thus, $q_z(\eta,\psi,\zeta,\ell)$ is only available for those circulation iterations where subroutine QLDS is called to compute a new QMAT and $q_z(\eta,\psi,\zeta,\ell)$.

The primary reason for computing and storing the SUM matrix

$q_z(\eta,\psi,\zeta,\ell)$ is to permit its print-out. This allows the user to determine the contribution of each individual rotor blade (ζ) and each type of wake (ℓ) to $\lambda(\eta,\psi)$. This is often very useful information and it is sometimes worth the extra expense to have QMAT recomputed on the last circulation iteration of a LDS-73 run, just to obtain $q_z(\eta,\psi,\zeta,\ell)$ for the final, converged $\gamma_{bv}(\phi)$.

The SUM matrix is quite large (typically $6\eta*25\psi*6\zeta*3\ell=$ 2700 words). Unlike QMAT, the SUM matrix is of no use once it has been printed out and $\lambda(\eta,\psi)$ has been computed. Thus, all of the large variable arrays associated with the blade motion and airloads computations can be stored in the same space as SUM. This greatly reduces the storage penalty associated with SUM.

3.2.3 Tip Vortex

Define an x, y, z coordinate system with origin at point P_η, where the induced velocity is to be computed. A set of unit vectors is defined $\vec{i}$, $\vec{j}$, $\vec{k}$ corresponding to x, y, z, respectively. The y axis and $\vec{j}$ are parallel to the tip path plane (TPP) and are directed downstream (Fig. 16). The x-axis and $\vec{i}$ are parallel to the TPP and are directed to starboard. The z-axis and $\vec{k}$ are perpendicular to the TPP and are directed downward, making a right-handed, rectangular coordinate system.

The various tip vortex models are described in Subsection 2.5. All of these models start from a tip vortex line made up of straight-line segments. The geometry of such a line segment is defined by the vectors $\vec{a}$ and $\vec{b}$ from point P_η to the ends of the segment (Fig. 16). The components of these vectors are:

$$\vec{a} = x_a\vec{i} + y_a\vec{j} + z_a\vec{k} \tag{3.9}$$

$$\vec{b} = x_b\vec{i} + y_b\vec{j} + z_b\vec{k} \tag{3.10}$$

To compute these conponents, recall that the advance ratio μ and the mean inflow λ are both normalized by rotor tip speed ΩR and that azimuth angles are measured in radians. Thus, the rotor hub will move upstream a distance $\mu\delta$ as the rotor rotates through azimuth angle δ. A review of the notation defined in Subsection 1.5 is suggested.

For the rigid-wake case, the horizontal components are readily found using Fig. 16:

$$x_a = \ell \sin\phi - \eta \sin\psi \tag{3.11}$$

$$x_b = \ell \sin(\phi - \Delta\phi) - \eta \sin\psi \tag{3.12}$$

$$y_a = \ell \cos\phi - \eta \cos\psi + \mu\delta \tag{3.13}$$

$$y_b = \ell \cos(\phi - \Delta\phi) - \eta \cos\psi + \mu(\delta + \Delta\phi) \tag{3.14}$$

where the age δ of point P_ℓ is:

$$\delta = \psi + \zeta - \phi \tag{1.3}$$

and $\ell = \ell_3$ for the tip vortex.

For the rigid-wake case, the wake moves downwards perpendicular to the TPP by $\lambda\delta$ as the rotor rotates through an angle δ, where λ is a mean inflow input to LDS-73 just for this purpose. Including the effect of zeroth harmonic flapping a_o, the vertical components are (for rigid wake):

$$z_a = \lambda\delta - a_o(\ell - \eta) \tag{3.15}$$

$$z_b = \lambda(\delta + \Delta\phi) - a_o(\ell - \eta) \tag{3.16}$$

where a_o is another input to LDS-73 and can be estimated using Eq. 3.3.

For the distorted wake case, the distortion $\vec{D}(\phi,\delta)$ must be included. The distortion is computed by WG-71 and input to LDS-73.

It has components:

$$\vec{D}(\phi,\delta) = D_1(\phi,\delta)\,\vec{i} + D_2(\phi,\delta)\,\vec{j} + D_3(\phi,\delta)\,\vec{k} \qquad (3.17)$$

The horizontal components of $\vec{a}$ and $\vec{b}$ become (for distorted wake):

$$x_a = \ell\sin\phi - \eta\sin\psi + D_1(\phi,\delta) \qquad (3.18)$$

$$x_b = \ell\sin(\phi-\Delta\phi) - \eta\sin\psi + D_1(\phi-\Delta\phi,\ \delta+\Delta\phi) \qquad (3.19)$$

$$y_a = \ell\cos\phi - \eta\cos\psi + \mu\delta + D_2(\phi,\delta) \qquad (3.20)$$

$$y_b = \ell\cos(\phi-\Delta\phi) - \eta\cos\psi + \mu(\delta+\Delta\phi) + D_2(\phi-\Delta\phi,\ \delta+\Delta\phi) \qquad (3.21)$$

For the vertical components of $\vec{a}$ and $\vec{b}$, recall that $\vec{D}(\phi,\delta)$ includes the effect of the induced part of λ but not the effect of the component of the relative wind perpendicular to the TPP ($\mu\tan i$):

$$z_a = \delta\mu\tan i - a_0(\ell-\eta) + D_3(\phi,\delta) \qquad (3.22)$$

$$z_b = (\delta+\Delta\phi)\mu\tan i - a_0(\ell-\eta) + D_3(\phi-\Delta\phi,\ \delta+\Delta\phi) \qquad (3.23)$$

where i is the angle between the TPP and the relative wind (Fig. 15) and is an input to LDS-73 (Eq. 3.4).

The circulation of the vortex-line segment must be known in addition to $\vec{a}$ and $\vec{b}$. The wake models of LDS-73 all use constant circulation trailing-wake segments. Thus, only γ_a, the normalized circulation at point P_a, is required. When the tip vortex is modeled by only a vortex line, then

$$\gamma_a = \gamma_{bv}(\phi) \qquad (2.10)$$

When the tip vortex sheet plus vortex-line model is used (Subsection 2.5.3),

$$\gamma_a = f_t(\phi, \delta) * \gamma_{bv}(\phi) \tag{2.14}$$

For the computation of $\gamma_{bv}(\phi)$, see Subsection 3.1.

The contribution of a straight vortex-line segment to the induced velocity at point P_η is derived in Appendix A as:

$$\vec{q}_{VL} = \gamma_a \left(\frac{b}{2R}\right)\left[\frac{(|\vec{a}| + |\vec{b}|)(|\vec{a}||\vec{b}| - \vec{a}\cdot\vec{b})}{|\vec{a}||\vec{b}|\,|\vec{a}\times\vec{b}|^2}\right](\vec{a}\times\vec{b}) \tag{A.24}$$

This q_{VL} is normalized by tip speed and is parallel to $\vec{a}\times\vec{b}$. It also neglects all vortex core and close blade-vortex interaction effects. Define the components of $\vec{a}\times\vec{b}$ as:

$$\vec{a}\times\vec{b} = (a\times b)_i\,\vec{i} + (a\times b)_j\,\vec{j} + (a\times b)_k\,\vec{k} \tag{3.24}$$

Define F_c as a correction factor which multiplies γ_a to correct for all vortex core and close blade-vortex interaction effects. Recall that QMAT is multiplied by $(b/2R)\gamma_{bv}(\phi)$ to get $\lambda(\eta,\psi)$; hence the $f_t(\phi,\delta)$ factor in γ_a must be included in QMAT. For the instantaneous tip vortex rollup case, where Eq. 2.10 is used for γ_a, the factor $f_t(\phi,\delta)$ is always equal to one. From Eq. 3.6, QMAT is a sum of $q_k(\eta,\psi,\zeta,\ell,\phi)$. From Subsection 3.2.1, $q_k(\eta,\psi,\zeta,\ell,\phi)$ is normalized by $\Gamma/4\pi R$ instead of ΩR. From the definition of γ_a,

$$\frac{\Gamma}{4\pi R} = \gamma_a\left(\frac{b}{2R}\right)\Omega R \tag{3.25}$$

Therefore, the tip vortex-line segment contribution is

$$q_k(\eta,\psi,\zeta,\ell_3,\phi) = F_c * f_t(\phi,\delta)\left[\frac{(|\vec{a}| + |\vec{b}|)(|\vec{a}||\vec{b}| - \vec{a}\cdot\vec{b})}{|\vec{a}||\vec{b}|\,|\vec{a}\times\vec{b}|^2}\right](a\times b)_k \tag{3.26}$$

Similar expressions can be written for the induced velocity components in the $\vec{i}$ and $\vec{j}$ directions. The calculation of F_c will be described in Subsection 3.2.4.

If the tip vortex sheet plus vortex line model is used, the contribution of the vortex-sheet segment must be added to the results of Eq. 3.26. Except for the near wake case, the geometry of the tip-vortex sheet is determined by translating the tip vortex line inboard, perpendicular to itself, from $\ell = 1.0$ to $\ell = \ell_t(\phi)$. The geometry is specified by four parameters $\vec{a}$, $\vec{b}$, $\Delta\tilde{x}$, and z_m. The $\vec{a}$ and $\vec{b}$ are the same as in the tip vortex-line case, see above. The length of the vortex sheet is:

$$\Delta\tilde{x} = 1.0 - \ell_t(\phi) \tag{3.27}$$

The z component of a vector from point P_η to the midpoint of the $\ell_t(\phi)$ edge of the vortex sheet is:

$$z_m = 0.5\,(z_a + z_b) - a_o\,\Delta\tilde{x} \tag{3.28}$$

where z_a and z_b are the z components of $\vec{a}$ and $\vec{b}$ (Eqs. 3.15 and 3.16 or 3.22 and 3.23) and a_o is the zeroth harmonic of rigid body flapping.

The computation of the contribution of a vortex sheet segment to the induced velocity at point P_η is complex. The numerous equations required are developed in Appendix B. The ultimate result ($\vec{q}_{vs}$) is normalized by ΩR. To get the contribution to $q_k(\eta,\psi,\zeta,\ell_3,\phi)$, take the $\vec{k}$ component of $\vec{q}_{vs}$, eliminate the $\gamma_a(b/2R)$ factor in front, and multiply by $[1 - f_t(\phi,\delta)]$. For example, if the $\vec{q}_{vs}$ of Eq. B.69 is used (the constant circulation, no vortex-core case), then

$$\Delta q_k(\eta,\psi,\zeta,\ell_3,\phi) = \left[1 - f_t(\phi,\delta)\right]\left(I_1\, z_x + I_2\, z_z\right) \tag{3.29}$$

The analogous expression including the vortex core is

$$\Delta q_k(\eta,\psi,\zeta,\ell_3,\phi) = \left[1 - f_t(\phi,\delta)\right]\left[\left(I_{1a} + I_{1b}\right) z_x + \left(I_{2a} + I_{2b}\right) z_z\right] \tag{3.30}$$

For the near-wake case, a special tip vortex-sheet geometry must be used (Subsection 2.5.3). The vortex-sheet segment is taken perpendicular to the rotor blade instead of to the tip vortex line. In the notation of Appendix B:

$$C_a = 0 \tag{3.31}$$

$$C_b = \Delta\phi \tag{3.32}$$

$$x_1 = 1.0 - \eta \tag{3.33}$$

$$x_2 = \ell_t(\phi) - \eta \tag{3.34}$$

$$z_\perp = 0 \tag{3.35}$$

and $\Delta\tilde{x}$ is given by Eq. 3.27. When $\ell_t(\phi) \leq \eta \leq 1.0$, there is a singularity, but it is integrable as long as η is not allowed to equal either 1.0 or $\ell_t(\phi)$. Due to C_a and z being equal to zero, the computations are greatly simplified and the results are given here. For the case of no vortex core ($\rho_{vs} = 0$):

$$\Delta q_{fk}(\eta,\psi,\zeta,\ell_3,\phi) = \left[1 - f_t(\phi,S)\right] \ln\left|\frac{x_2\left(\sqrt{C_b^2 + x_1^2} - C_b\right)}{x_1\left(\sqrt{C_b^2 + x_2^2} - C_b\right)}\right| \frac{1}{\Delta\tilde{x}} \tag{3.36}$$

When $C_b < \rho_{vs}$:

$$\Delta q_{fk}(\eta,\psi,\zeta,\ell_3,\phi) = \frac{C_b\left[1 - f_t(\phi,S)\right]}{\Delta\tilde{x}\sqrt{\rho_{vs}^2 - C_b^2}} \left[\tan^{-1}\left(\frac{\sqrt{C_b^2 + x_1^2}}{\sqrt{\rho_{vs}^2 - C_b^2}}\right) - \tan^{-1}\left(\frac{\sqrt{C_b^2 + x_2^2}}{\sqrt{\rho_{vs}^2 - C_b^2}}\right)\right] \tag{3.37}$$

When $C_b > \rho_{vs}$:

$$\Delta q_{k}(\eta,\psi,\zeta,\ell_3,\phi) = \frac{C_b[1-f_t(\phi,\zeta)]}{2\Delta\tilde{x}\sqrt{C_b^2-\rho_{vs}^2}} \ln\left[\frac{\left(\sqrt{C_b^2+x_1^2}-\sqrt{C_b^2-\rho_{vs}^2}\right)\left(\sqrt{C_b^2+x_2^2}+\sqrt{C_b^2-\rho_{vs}^2}\right)}{\left(\sqrt{C_b^2+x_1^2}+\sqrt{C_b^2-\rho_{vs}^2}\right)\left(\sqrt{C_b^2+x_2^2}-\sqrt{C_b^2-\rho_{vs}^2}\right)}\right] \tag{3.38}$$

3.2.4 Tip Vortex Core and Close Blade-Vortex Interaction Effects

The tip vortex correction factor F_c is used to include all vortex core and close blade-vortex interaction effects on the induced velocity $q_k(\eta,\psi,\zeta,\ell,\phi)$. The correction factor F_c is the product of the vortex core factor $\gamma_c(\rho)$ and the close blade-vortex interaction factor q_f:

$$F_c = q_f * \gamma_c(\rho) \tag{3.39}$$

The vortex core factor $\gamma_c(\rho)$ is the fraction of the total tip vortex-line segment circulation γ_a which is within a distance ρ of the center of the vortex line. For large ρ this fraction $\gamma_c(\rho)$ goes to 1.0 and can be neglected. The function $\gamma_c(\rho)$ depends on the choice of vortex-core model (Subsection 2.4). The normal choice is the fixed-wing vortex-core model,for which:

$$\gamma_c(\rho) = \frac{\rho^2}{1+\rho^2} \tag{2.3}$$

The distance ρ is normalized by the vortex-core radius ρ_c. The length of a perpendicular dropped from point P_η to the vortex-line segment (or its extension) is h (normalized by R). From Eqs. A.19 and A.20 in Appendix A:

$$h = \frac{|\vec{a} \times \vec{b}|}{|\vec{c}|} \tag{3.40}$$

Recalling that ρ_c is normalized by R:

$$\rho = h/\rho_c = \frac{|\vec{a} \times \vec{b}|}{|\vec{c}|\,\rho_c} \tag{3.41}$$

For the fixed-wing vortex-core model:

$$\gamma_c(\rho) = \frac{h^2}{h^2+\rho_c^2} = \frac{|\vec{a}\times\vec{b}|^2}{|\vec{a}\times\vec{b}|^2 + |\vec{c}|^2\rho_c^2} \tag{3.41}$$

Putting this into Eq. 3.26:

$$q_k(\eta,\psi,\varsigma,l,\phi) = q_f\left[f_t(\phi,\varsigma)\right]\left[\frac{(|\vec{a}|+|\vec{b}|)(|\vec{a}||\vec{b}| - \vec{a}\cdot\vec{b})}{|\vec{a}||\vec{b}|(|\vec{a}\times\vec{b}|^2+|\vec{c}|^2\rho_c^2)}\right](a\times b)_k \tag{3.42}$$

For the other vortex-core models, Eq. 3.41 is used to provide ρ for the appropriate $\gamma_c(\rho)$ function. For large ρ (typically, $\rho > 10^4$) $\gamma_c(\rho)$ can be set equal to 1.0 to save computation.

A close blade-vortex interaction can change F_c in two ways. It can cause tip vortex bursting, which enlarges ρ_c and hence changes $\gamma_c(\rho)$, or it can change q_f. The first step is to determine whether or not a particular tip vortex-line segment has had a close blade-vortex interaction. Consider a top view (projected into the TPP) of the tip vortex-line segment (with circulation Γ) and the η blade (Fig. 17). Define point P_c as the point on the vortex-line segment (or its extension) which intersects the η blade (or its extension) in this TPP projection. Let η_c be the radial station on the η blade of point P_c. The first requirement for a close blade-vortex interaction is that the point P_c lies on the η blade in the TPP projection. Assuming a 10% blade root cutout and providing a small margin at the tip, this means that η_c must be

$$0.10 \leq \eta_c \leq 1.01 \tag{3.43}$$

The magnitude of η_c can be found with the use of Fig. 17 and some manipulation:

$$\eta_c = \frac{x_a y_b - x_b y_a}{\Delta y \sin\psi - \Delta x \cos\psi} + \eta \tag{3.44}$$

where:

$$\Delta x = x_b - x_a \tag{3.45}$$

$$\Delta y = y_b - y_a \tag{3.46}$$

Notice that the numerator is equal to $(axb)_k$, which must be calculated in any case as part of Eq. 3.26.

Define the distance from point P_c to point P_a in the TPP projection as $\Delta\ell_{xy}$ (normalized by vortex-line segment length). Point P_a is at the end of the tip vortex-line segment which has that largest azimuth angle ϕ. The sign of $\Delta\ell_{xy}$ is positive when point P_a is behind the η blade, as shown in Fig. 17:

$$\Delta\ell_{xy} = \frac{y_a - \Delta\eta\cos\psi}{\Delta y} = \frac{x_a - \Delta\eta\sin\psi}{\Delta x} \tag{3.47}$$

where:

$$\Delta\eta = \eta_c - \eta \tag{3.48}$$

Two versions of the $\Delta\ell_{xy}$ formula are given because either Δx or Δy can be zero.

The second requirement for a close blade-vortex interaction is that the nearest end of the tip vortex-line segment be no more than one segment length behind the η blade ($\Delta\ell_{xy} < 1.0$) or more than one-half the segment length ahead of the η blade ($\Delta\ell_{xy} > -1.5$).

$$-1.5 < \Delta\ell_{xy} < 1.0 \tag{3.49}$$

This is a generous definition and could be restricted further if desired.

The two requirements outlined above are sufficient to require the use of the factor q_f to represent the close blade-vortex interaction; the penalty for computing the factor q_f when it is not really needed is extra computation, not incorrect results. For the vortex bursting case, more stringent criteria must be met because bursting at the wrong time gives incorrect results. The first requirement remains, as stated in Eq. 3.43. The second requirement is restricted further in the area ahead of the η blade, since bursting is expected to occur at or behind the η blade:

$$DVB < \Delta \ell_{xy} < 1.0 \tag{3.50}$$

where DVB is an input to LDS-73. A typical value for DVB is zero. This implies that all of the tip vortex-line segment is behind the η blade.

The bursting model requires that all of a tip vortex-line segment be burst or none of it. This tends to cause the choice of a smaller DVB than would otherwise the used. For example, suppose that the theory says the tip vortex-line should burst at $\Delta\ell_{xy}$ = 0.10 and that point P_a is at $\Delta\ell_{xy}$ = 0.05. Ideally, 95% of the segment should be burst and 5% not. A DVB = 0.10 would cause no bursting while a DVB = 0.0 would cause 100% bursting. The latter choice is probably best, pending a more complex model which allows partial bursting of a segment.

This leaves the case where part of the tip vortex-line segment has passed the η blade, but not enough to make $\Delta\ell_{xy}$ > DVB. This segment will not burst immediately, but will burst later on. To provide for this case, the age at which the segment bursts $\delta_b(\phi)$ is increased by $\Delta\phi$ for every increment of 0.5 (or fraction thereof), that $\Delta\ell_{xy}$ is less than DVB.

A third requirement for tip vortex-line segment bursting is that the segment (or its extension) pass within a distance FVB*ρ_c below (or above) the η blade. Define z_c as the distance of point P_c (on the tip vortex-line segment or its extension) from the η blade (Fig. 18). Since point P_c lies on the η blade in the TPP projection (Fig. 17), z_c is a distance perpendicular to the TPP. From Fig. 18:

$$z_c = z_a + a_0 \Delta\eta - \Delta\ell_{xy}(z_b - z_a) \tag{3.51}$$

Now the third requirement can be written:

$$FVB \geqslant |z_c/\rho_c| \tag{3.52}$$

The absolute value is necessary because the segment can be either above or below the η blade.

To summarize the tip vortex core bursting criteria, consider a tip vortex-line segment identified by azimuth angle ϕ and age δ. If the basic criteria of Eqs. 3.43, 3.50, and 3.52 are all met, then bursting occurs immediately:

$$\delta_b(\phi) = \delta \tag{3.53}$$

If Eq. 3.50 is not satisfied, but Eq. 3.49 is, then

$$\delta_b(\phi) = \begin{cases} \delta + \Delta\phi & \text{for } 0 < (DVB - \Delta l_{xy}) < 0.5 \\ \delta + 2\Delta\phi & \text{for } 0.5 < (DVB - \Delta l_{xy}) < 1.0 \\ \delta + 3\Delta\phi & \text{for } 1.0 < (DVB - \Delta l_{xy}) < 1.5 \end{cases} \tag{3.54}$$

Assuming that Eqs. 3.43 and 3.49 are satisfied for a particular tip vortex-line segment, it becomes necessary to compute q_f. This q_f represents the effects of the extra near wake generated by a close blade-vortex interaction. The effects are estimated using a lifting surface solution by Johnson [52,53], see Subsection 2.5.2. The expression for q_f used in LDS-73 is a simplified one, specialized to Mach number (M) of zero. For a more general $M \neq 0$ case, the use of Johnson's Eq. 40 [53] is suggested. This expression is very complex and will not be repeated here. However, the necessary data to use this expression and the relationship of Johnson's notation to the notation used here will be given.

The form of Johnson's Eq. 40 is

$$\frac{L\sqrt{1-M^2}}{2\pi\rho V^2 b} = \frac{\Gamma}{2\pi V b}\left\{ - \right\} \tag{3.55}$$

To get q_f, the lift (L) given by Eq. 3.55 must be divided by the lift (L*), computed neglecting lifting-surface theory or the near wake. Consider an infinite, straight vortex line of circulation Γ passing below the η blade, parallel to the TPP, at a distance z_c and making an angle Λ with the η blade (Fig. 19). A portion of this infinite vortex line is designated the vortex-line segment and has a Δx and a Δy just like the segment of Fig. 17. The definitions of ψ, $\Delta\eta$, point P_η, and point P_c in Fig. 19 are the same as in Fig. 17. Now, using the Biot-Savart relation for an infinite, straight vortex line, the z component of the induced velocity at point P_η is

$$q_k^* = \frac{\Gamma}{4\pi R}\left[\frac{2\,\Delta\eta\cos\delta}{z_c^2+(\Delta\eta\cos\delta)^2}\right] \tag{3.56}$$

Assuming that the angle of attack α is a small angle,

$$L^* = \frac{1}{2}\rho V^2 c\,\frac{a}{\sqrt{1-M^2}}\,\alpha = \rho V b\,\frac{a}{\sqrt{1-M^2}}\,q_k^* \tag{3.57}$$

where b is the semi-chord (c/2) and $a/\sqrt{1-M^2}$ is the lift curve slope $(\partial C_L/\partial\alpha)$. Now

$$\frac{L^*\sqrt{1-M^2}}{2\pi\rho V^2 b} = \left(\frac{a}{2\pi}\right)\left(\frac{b}{R}\right)\frac{\Gamma}{2\pi V b}\left[\frac{\Delta\eta\cos\delta}{z_c^2+(\Delta\eta\cos\delta)^2}\right] \tag{3.58}$$

Dividing Eq. 3.55 by Eq. 3.58 to obtain $q_f = L/L^*$,

$$q_f = \left(\frac{2\pi}{a}\right)\left(\frac{R}{b}\right)\left[\frac{z_c^2+(\Delta\eta\cos\delta)^2}{\Delta\eta\cos\delta}\right]\left\{-\right\} \tag{3.59}$$

where the expression {-} is the {-} expression in Johnson's Eq. 40 [53].

To compute the {-} expression, the following inputs are needed (in Johnson's notation): r^A, h, Λ. The lengths r^A and h are both normalized by the semi-chord (b). In terms of the notation used here:

$$r^R = -\Delta\eta\,(R/b) \tag{3.60}$$

$$h = |z_c|\,(R/b) \tag{3.61}$$

$$\Lambda = \frac{\pi}{2} + \delta \tag{3.62}$$

Now Λ and δ can be calculated (Fig. 19):

$$\sin\Lambda = \cos\delta = \frac{\Delta y \sin\psi - \Delta x \cos\psi}{\sqrt{\Delta x^2 + \Delta y^2}} \tag{3.63}$$

$$\cos\Lambda = -\sin\delta = -\frac{\Delta y \cos\psi + \Delta x \sin\psi}{\sqrt{\Delta x^2 + \Delta y^2}} \tag{3.64}$$

This completes the computation of q_f and of F_c.

3.2.5 Inboard Trailing Wake

The computation of inboard-trailing-wake induced velocities is similar to that for the tip vortex but much simpler. Since the inboard trailing wake does not involve a concentrated vortex line, such as the tip vortex, the use of a rigid-wake geometry is adequate. Also, the close blade-vortex interaction effects of vortex core bursting and of the lifting-surface solution can be neglected. Thus, the inboard trailing wake is modeled by a rigid wake vortex line, with an artificially enlarged vortex core, or by a rigid wake vortex sheet, see Subsection 2.6.

The x,y,z coordinate system defined for the tip vortex in Subsection 3.2.3 is also used here. The $\vec{a}$ and $\vec{b}$ definitions for the rigid wake tip vortex (Eqs. 3.9 to 3.16) also apply with an appropriate choice of ℓ. For a vortex line model ℓ is ℓ_2 (an input) and for a vortex sheet model ℓ is $\ell_i(\phi)$ (also an input). This definition of $\vec{a}$ and $\vec{b}$ means that the circulation γ_a of an inboard trailing-wake segment must be the negative of the γ_a for the corresponding tip-vortex segment, see Subsection 2.8.

For a vortex-line model, the vortex-core radius ρ_i is chosen to be about half the width of the average inboard trailing vortex sheet. This is done through the choice of ℓ_2 since:

$$\rho_i = \ell_2 - \ell_1 \tag{3.65}$$

Now, $q_k(\eta,\psi,\zeta,\ell,\phi)$ can be computed from Eq. 3.26. For this case, F_c is $\gamma_c(\rho)$, which is normally given by Eq. 2.3 (with $\rho_c = \rho_i$), and $f_t(\phi,\delta)$ is replaced by (-1.0) which represents the minus sign on γ_a discussed above.

For a vortex-sheet model, a vortex-core radius ρ_{vs} can be used, if desired. The length of the vortex-sheet segment ($\Delta\tilde{x}$) is:

$$\Delta\tilde{x} = \ell_i(\phi) - \ell_1 \tag{3.66}$$

The z coordinate of the midpoint of the back edge of the vortex-sheet segment (z_m) is the average of z_a and z_b computed from Eqs. 3.15 and 3.16 with $\ell = \ell_1$:

$$z_m = \lambda(\delta + \Delta\phi/2) - a_0(\ell_1 - \eta) \tag{3.67}$$

Now $q_k(\eta,\psi,\zeta,\ell,\phi)$can be computed in the same way as for the tip vortex sheet case but with the $[1 - f_t(\phi,\delta)]$ factor replaced by minus one to represent the minus sign of γ_a. The analog of Eq. 3.29 for the inboard trailing vortex sheet instead of the tip vortex is

$$q_k(\eta,\psi,\zeta,\ell_2,\phi) = -(I_1 z_x + I_2 z_z) \tag{3.68}$$

The near inboard trailing vortex sheet case must be handled specially in the same way that the near tip vortex sheet is handled. For this case:

$$C_b = \Delta\phi\, \ell_i(\phi) \tag{3.69}$$

$$X_1 = \ell_i(\phi) - \eta \tag{3.70}$$

$$X_2 = \ell_1 - \eta \tag{3.71}$$

Now $q_k(\eta,\psi,\zeta,\ \ell_2,\phi)$ is computed using Eqs. 3.36, 3.37, or 3.38 as necessary, but with the $[1 - f_t(\phi,\delta)]$ factor replaced by (-1.0).

3.2.6 Shed Wake

The computation of shed-wake induced velocities is similar, in principle, to the trailing wake, but differs in detail. Like the inboard trailing wake, there is no concentrated vortex line in the shed wake and the effects of vortex core bursting and of the lifting surface solution can be neglected. Again, the rigid-wake geometry is adequate; however, as a matter of convenience, the outboard end of the shed wake is located on the tip vortex, even when the tip vortex is distorted. For airloads computations in LDS-73,only the vortex-line shed-wake model is used. Vortex-sheet, shed-wake models have been tried in earlier programs and the results were very similar to those of the vortex-line model at greatly increased expense.

The standard x,y,x coordinate system is used (see Subsection 3.2.3). Point P_a is at the outboard end of the shed-vortex-line segment and point P_b is at the inboard end. To save extra computation, the $\vec{a}$ for the shed-vortex line is the same as the $\vec{a}$ for the tip vortex line (Eqs. 3.11, 3.13, 3.15 or 3.18, 3.20, 3.22). If the distorted-wake model is used for the tip vortex, this means that the outboard end of each shed-wake segment will be located by the distorted wake model. The inboard end of each shed-wake segment is always located by the rigid wake model with $\ell = \ell_2$:

$$x_b = \ell_2 \sin\phi - \eta \sin\psi \tag{3.72}$$

$$y_b = \ell_2 \cos\phi - \eta \cos\psi + \mu\delta \tag{3.73}$$

$$z_b = \lambda\delta - a_0(\ell_2 - \eta) \tag{3.74}$$

The circulation of each shed-wake vortex-line segment is $\gamma_s(\phi)$, (Eq. 2.9). The vortex-core radius ρ_s is chosen to approximately fill the space between adjacent shed-vortex-line segments, thus approximating a continuous vortex sheet:

$$\rho_s = 0.4\ \Delta\phi \tag{3.75}$$

For the shed wake the correction factor F_c is simply $\gamma_c(\rho)$, which is normally given by Eq. 2.3 (with $\rho_c = \rho_s$). The factor $f_t(\phi,\delta)$ is always one for the shed wake. Recall that the shed wake is identified by ℓ_1; therefore $q_k(\eta,\psi,\zeta,\ell_1,\phi)$ is computed from Eq. 3.26 with the appropriate F_c, $f_t(\phi,\delta)$, $\vec{a}$, and $\vec{b}$.

The near shed wake is normally represented by a vortex sheet. This vortex sheet starts a distance ε_s behind the points P_η on the η blade and extends over the first $\Delta\phi$ interval (see Subsection 2.7). Recall the definition of the shed circulation γ_s (ϕ):

$$\gamma_s(\phi) = \gamma_{bv}(\phi+\Delta\phi) - \gamma_{bv}(\phi) \tag{2.9}$$

.

For the far shed wake, all of the wake shed between ϕ and $\phi + \Delta\phi$ is accumulated into a shed-vortex line at ϕ. For the near shed wake, it is spread out in a sheet between $\phi + \Delta\phi$ and ϕ. The geometry of the sheet is referred to the η blade which is at $\psi = \phi + \Delta\phi$. This is awkward because $\vec{a}$ and $\vec{b}$ are defined at ϕ for use in the trailing wake computations, but not at $\phi + \Delta\phi$. To avoid defining an extra set of $\vec{a}$ and $\vec{b}$ at $\phi + \Delta\phi$, the computation of the near shed-wake segment is in LDS-73 deferred one $\Delta\phi$-step until $\psi = \phi$. Thus, $\gamma_s(\phi - \Delta\phi)$ must be used for the circulation of the near shed-wake segment. This allows the far shed-wake equations for $\vec{a}$ and $\vec{b}$ to be used for the near shed wake, with an extra term to include the effect of ε_s. For x_a and x_b, subtract $\varepsilon_s \cos\psi$. For y_a and y_b, add $\varepsilon_s \sin\psi$. For z_a and z_b, the age δ is equal to ε_s.

In addition to $\vec{a}$ and $\vec{b}$, the vortex-sheet computation requires $\Delta\tilde{x}$ and z_m. The length of the vortex-sheet segment ($\Delta\tilde{x}$) is defined by the distance between adjacent shed vortex lines, at their midpoints, with a correction for ε_s:

$$\Delta\breve{x} = \frac{\Delta\phi}{2}(l_3 - l_2) - \varepsilon_s \qquad (3.76)$$

The z coordinate of the midpoint of the back edge of the sheet segment (z_m) is the average of z_a and z_b for $\phi - \Delta\phi$. Now, $q_k(\eta,\psi,\zeta,\ell,\phi)$ can be computed in the same way as for the tip vortex sheet, but with $[1 - f_t(\phi,\delta)]$ replaced by plus one. The analog of Eq. 3.29 for the near shed wake is

$$q_k(\eta,\psi,\zeta,\ell_1,\phi) = (I_1 z_x + I_2 z_z) \qquad (3.77)$$

3.2.7 Special Cases

There are three special cases arising during induced-velocity computations and not covered elsewhere. They are: avoidance of the singularity at the edge of a vortex sheet, limitation of the maximum angle-of-attack change ($\Delta\alpha_m$), due to any wake element, and the consideration of tangential induced velocities.

It is shown in Appendix B (Subsection B.1) that there is a singularity when point P_η approaches the edges of a vortex-sheet segment (i.e., when $x_1 \to 0$ or $x_2 \to 0$). This singularity is a result of the vortex-sheet segment model where the circulation drops sharply from a finite value to zero as a step function at the edges. In reality the circulation would either taper off to zero gradually or would continue, without a break, into the adjacent segment. Either way there is no singularity in reality. A more accurate vortex sheet model of reality would therefore avoid the singularity. This would be very complicated and expensive, however, and the simpler solution of avoiding the edges is adopted.

Point P_η is accurately fixed on the rotor blade. On the other hand, the edges of the vortex-sheet segments, with the exception of the near-wake case, are located rather arbitrarily by the requirements of the plane, rectangular, vortex-sheet-segment model. Thus, these edges can be moved to avoid the singularity without particularly changing the accuracy of the model. In LDS-73, the values of

x_1^2 and x_2^2 are checked to see if they are greater than $(\text{VSM}*\Delta\tilde{x})^2$, where the vortex-sheet margin (VSM) is an input (typically 0.1 or 0.2). If either x_1 or x_2 is not large enough, then it is increased to $\text{VSM}*\Delta\tilde{x}$.

As a result of some investigations by Ham [54,55], it is postulated that there is a maximum incremental lift coefficient $\Delta C_{L_{max}}$, due to any blade-vortex interaction of 0.2 to 0.3. This corresponds to a maximum change in angle of attack ($\Delta\alpha_{max}$) of 0.035 to 0.052 radians. The LDS-73 program does not isolate each individual blade-vortex interaction but instead considers one vortex line (or sheet) segment at a time. Fortunatley, in most cases, the major portion of the induced velocity contribution of a blade-vortex interaction comes from one tip vortex-line segment. The change in angle of attack at point P_η ($\Delta\alpha$), due to each tip vortex-line segment is readily calculated. Assuming that $\Delta\alpha$ is a small angle,

$$\Delta\alpha = \gamma_a \left(\frac{b}{2R}\right) q_k(\eta,\psi,\zeta,\ell_3,\phi)/U_T(\eta,\psi) \qquad (3.78)$$

where $U_T(\eta,\psi)$ is the tangential component of velocity at point P_η (Fig. 20). If $\Delta\alpha$ is greater than $\Delta\alpha_{max}$, then $q_k(\eta,\psi,\zeta,\ell_3,\phi)$ is multiplied by $(\Delta\alpha_{max}/\Delta\alpha)$ which reduces $\Delta\alpha$ to $\Delta\alpha_{max}$. Since $\Delta\alpha_{max}$ is an input to LDS-73,various values can be tested.

Normally, only the z component (perpendicular to the TPP) of the induced velocity (q_k) is considered in airloads computations. This is because the induced velocity is frequently an important part of the total downwash $\lambda(\eta,\psi)$. On the other hand, the induced contribution (q_T) to the tangential velocity $U_T(\eta,\psi)$ is normally assumed to be small compared to the contribution of the helicopter airspeed and the rotor rotation ($\eta + \mu\sin\psi$). To test the importance of q_T, provision to compute it has been made in LDS-73. This is easily done using q_k and the components of $\vec{a} \times \vec{b}$, which is a vector parallel to the induced velocity (Appendix A, Eq. A.22):

$$q_t(\eta,\psi,\zeta,l,\phi) = q_i \sin\psi - q_j \cos\psi \tag{3.79}$$

$$q_i = q_k(\eta,\psi,\zeta,l,\phi) * (a\times b)_i \,/ (a\times b)_k \tag{3.80}$$

$$q_j = q_k(\eta,\psi,\zeta,l,\phi) * (a\times b)_j \,/ (a\times b)_k \tag{3.81}$$

To save computation time, a QTAN(η,ψ,ϕ) matrix is defined which is analogous to the QMAT(η,ψ,ϕ) matrix:

$$QTAN(\eta,\psi,\phi) = \sum_{\zeta} \sum_{n=0}^{m-1} \Bigg[\sum_{l=l_2}^{l_3} q_t(\eta,\psi,\zeta,l,\phi+2\pi n) + q_t(\eta,\psi,\zeta,l_1,\phi-\Delta\phi+2\pi n) - q_t(\eta,\psi,\zeta,l_1,\phi+2\pi n) \Bigg] \tag{3.82}$$

The total tangential component of induced velocity at point P_η is obtained by summing QTAN(η,ψ,ϕ) over ϕ:

$$q_T(\eta,\psi) = \left(\frac{b}{2R}\right) \sum_{\phi} \left[\gamma_{bv}(\phi) * QTAN(\eta,\psi,\phi) \right] \tag{3.83}$$

Finally,

$$U_T(\eta,\psi) = \eta + \mu \sin\psi + q_T(\eta,\psi) \tag{3.84}$$

In LDS-73, both $q_T(\eta,\psi)$ and $(\eta + \mu\sin\psi)$ are always computed and printed out side-by-side for comparison. There are three options in the airloads computation:

1) $q_T(\eta,\psi)$ is neglected in $U_T(\eta,\psi)$
2) $q_T(\eta,\psi)$ is included in $U_T(\eta,\psi)$
3) two sets of airloads are computed using (1) and (2) for comparison.

Normally option (1) is used because $q_T(\eta,\psi)$ is not very important.

3.3 Blade Motion

Blade motion is only important in helicopter rotor airloads computations to the extent that it influences the airloads. Blade motion influences the airloads when it is of large enough amplitude to change the local angle of attack of a rotor-blade section. At low advance ratios the modes of blade motion which have a significant effect on airloads are rigid-body flapping and first mode flapwise bending. At higher advance ratios, blade torsional motion can also become important. Since the work reported here is primarily concerned with low advance ratios, where tip-vortex distortion is important, blade-torsional motion is neglected. For a more general airloads computation, torsional motion should be included.

If blade stresses are to be computed, many more modes of blade motion need to be considered. It is suggested that this is best done by developing a separate dynamics program which includes a complete blade motion analysis but which uses airlaods computed by a separate program similar to LDS-73. This decoupling of the small amplitude blade motion from the airloads saves a substantial amount of computation. No such dynamics program has been developed to go with LDS-73, since blade stresses are not the object of the work reported here.

Two different solutions for the blade motion are used in LDS-73. The harmonic solution considers only the zeroth, first, and second harmonics of rigid-body flapping and neglects blade bending. The numerical solution includes all harmonics of rigid-body flapping and of the first mode of flapwise blade bending. The harmonic solution is much simpler and faster and it is used to provide a starting point for the iteration required by the more accurate numerical solution. The harmonic solution is also useful because it is possible to isolate the various contributions to the flapping motion from: thrust, blade twist, and the individual downwash harmonics. This data can provide a useful insight into the sources of error in the computed flapping motion.

Rotor trim is specified by thrust coefficient (C_T) and TPP incidence (i) in both solutions. The thrust (T) is assumed to act perpendicular to the tip-path plane (TPP) and to be the vector sum of the weight (W) and the propulsive force (X):

$$T = \sqrt{W^2 + X^2} \tag{3.85}$$

The incidence of the TPP to the relative wind i (Fig. 15) is:

$$i = \sin^{-1}(X/T) \tag{3.4}$$

The LDS-73 program computes the horizontal force coefficient (C_H) so that it can be compared to C_T. As long as C_H is much less than C_T, the assumption that the thrust is perpendicular to the TPP is valid.

Other rotor trim parameters such as collective pitch and the flapping harmonics are computed by requiring equilibrium at the flapping hinge and by requiring that the input C_T and i be obtained. This approach gives the orientation of the rotor with respect to the relative wind, but it ignores the shaft and the fuselage. This is satisfactory because the orientation of the shaft and the fuselage are not needed in rotor airloads computations.

The harmonic solution for blade motion assumes that the lift-curve slope $\partial C_L/\partial\alpha$ is constant. This means that the compressibility correction to the lift can only be included in an average way, based on conditions at the 75% radius, and that stall is neglected. In the numerical solution, $\partial C_L/\partial\alpha$ is not held constant so compressibility and static-stall effects can be included. Dynamic-stall effects are not properly represented in the numerical solution, due to the difficulty of accurately estimating $\dot{\alpha} = \partial\alpha/\partial\psi$. Dynamic stall could be included in a completely revised version of the numerical solution, if necessary.

It must be emphasized that the primary importance of blade motion in airloads computations is the requirement for flapping equilibrium. This requirement causes a substantial redistribution of the rotor airloads and its importance increases with advance ratio (μ). Any forward flight airloads computation should include enough of a blade-motion solution to satisfy flapping equilibrium. Any further refinements in the blade-motion solution depend on the accuracy required and the flight conditions considered; i.e., high advance ratios require more sophisticated blade motion solutions.

3.3.1 Harmonic Solution

The fundamental assumption of the harmonic solution is that the blade motion consists of zeroth, first, and second harmonic rigid-body flapping:

$$\beta(\psi) = a_0 - a_1 \cos\psi - b_1 \sin\psi - a_2 \cos 2\psi - b_2 \sin 2\psi \qquad (3.86)$$

The downwash $\lambda(\eta,\psi)$, which was computed in Subsection 3.2, is also represented by its harmonics:

$$\begin{aligned} \lambda(\eta,\psi) = {} & \lambda_0(\eta) + \lambda_{1c}(\eta)\cos\psi + \lambda_{1s}(\eta)\sin\psi \\ & + \lambda_{2c}(\eta)\cos 2\psi + \lambda_{2s}(\eta)\sin 2\psi + \cdots \end{aligned} \qquad (3.87)$$

Using these expressions for β and λ while requiring flapping equilibrium and the achievement of a given C_T leads to a set of algebraic equations for the harmonics of β and the collective pitch θ_o.

The equation for moment equilibrium about an offset flapping hinge is:

$$I_b \frac{d^2\beta}{dt^2} + I_b \Omega^2 \beta = R^2 \int_{\ell_1}^{1.0} (\eta - \varepsilon)\, L(\eta,\psi)\, d\eta \qquad (3.88)$$

where I_b is the rotor blade flapwise moment of inertia about the flapping hinge, $L(\eta,\psi)$ is the airload per unit span at point P_η, and $(\eta-\varepsilon)$ is the moment arm of $L(\eta,\psi)$ about the flapping hinge. Figure 21 clarifies the definition of the flapping hinge offset ε (normalized by R).

The airload per unit span at point P_η can be written:

$$L(\eta,\psi) = \frac{1}{2}\rho a c(\Omega R)^2\left[F U_T^2 \alpha + \left(\frac{1-F}{2\pi}\right)\int_0^{2\pi} U_T^2 \alpha \, d\psi\right] \tag{3.89}$$

The lift deficiency function F is a alternate way of modeling the shed wake and is normally not used (F=1.0). The second term in Eq. 3.89 is because F does not apply to the steady-state component of $L(\eta,\psi)$. The tangential and perpendicular components of the airspeed U in the blade section plane at point P_η are U_T and U_P, respectively when normalized by tip speed (Fig. 20). For helicopters U_T is normally either much greater than U_P or U is so small that the dynamic pressure is too small to cause significant airloads. Therefore U_P^2 is neglected compared to U_T^2 in the harmonic solution. This is why U_T^2 is used instead of U^2 for the dynamic pressure in Eq. 3.89.

Before substituting Eq. 3.89 into Eq. 3.88, it is helpful to normalize. Define Lock Number (LN) as:

$$LN \equiv \frac{2\pi\rho c R^4}{I_b} \tag{3.90}$$

Instead of time t use azimuth angle ψ and define:

$$\dot{\beta} = \frac{d\beta}{d\psi} = \frac{1}{\Omega}\,\frac{d\beta}{dt} \tag{3.91}$$

$$\ddot{\beta} = \frac{d^2\beta}{d\psi^2} = \left(\frac{1}{\Omega}\right)^2 \frac{d^2\beta}{dt^2} \tag{3.92}$$

Now substituting into Eq. 3.88 and dividing by $I_b\Omega^2$:

$$\ddot{\beta} + \beta = \left(\frac{a}{2\pi}\right)\left(\frac{LN}{2}\right)\int_{\ell_1}^{1.0}\left[F U_T^2 \alpha + \left(\frac{1-F}{2\pi}\right)\int_0^{2\pi} U_T^2 \alpha \, d\psi\right] d\eta \tag{3.93}$$

This is the equation requiring moment equilibrium about the flapping hinge. The requirement for a given thrust coefficient C_T is obtained by integrating the airloads over the rotor:

$$C_T = \frac{\sigma a}{4\pi} \int_0^{2\pi} \int_{\ell_1}^{1.0} U_T^2 \alpha \, d\eta \, d\psi \tag{3.94}$$

where σ is the rotor solidity. These equations (3.93 and 3.94) can be solved for the flapping $\beta(\psi)$ and the collective pitch θ_o. Before this can be done, expressions for $U_T(\eta,\psi)$ and $\alpha(\eta,\psi)$ must be developed.

The angle of attack $\alpha(\eta,\psi)$ can be written (Fig. 20):

$$\alpha(\eta,\psi) = \theta(\eta) - \tan^{-1}(U_p/U_T) \tag{3.95}$$

The pitch $\theta(\eta)$ is not a function of azimuth ψ because the blade motion solution is done in the no-feathering plane (NFP), which eliminates cyclic pitch, and because blade-torsional motion is neglected. For the harmonic solution, a linear twist is assumed:

$$\theta(\eta) = \theta_o + \theta_1 \eta \tag{3.96}$$

Again, neglecting U_P^2 compared to U_T^2, the angle of attack can be written:

$$\alpha(\eta,\psi) = \theta_o + \theta_1 \eta - U_p/U_T \tag{3.97}$$

For the harmonic solution, the induced contribution to the tangential airspeed is neglected. Therefore:

$$U_T(\eta,\psi) = \eta + \mu \sin\psi \tag{3.98}$$

Assuming that the flapping β is a small angle:

$$U_p = \lambda(\eta,\psi) + \mu a_1 + \beta\mu\cos\psi + (\eta - \varepsilon)\dot{\beta} \tag{3.99}$$

The μa_1 term is the increment to $\lambda(\eta,\psi)$ caused by going from the TPP to the NFP and is analogous to the $\mu\tan i$ component of $\lambda(\eta,\psi)$, see Fig. 15.

All of the elements needed to solve Eqs. 3.93 and 3.94 are now available. The procedure starts with the development of an expression for $U_T^2\alpha$. Substituting Eq. 3.97:

$$U_T^2\alpha = U_T^2(\theta_0 + \theta_1\eta) - U_T U_P \tag{3.100}$$

Substituting Eq. 3.98 for U_T and Eq. 3.99 for U_P into Eq. 3.100:

$$U_T^2\alpha = (\theta_0 + \theta_1\eta)(\eta + \mu\sin\psi)^2 - (\eta + \mu\sin\psi)\left[\lambda(\eta,\psi) + \mu a_1 + \beta\mu\cos\psi + (\eta - \varepsilon)\dot{\beta}\right] \tag{3.101}$$

Now, Eq. 3.86 can be differentiated to obtain $\dot{\beta}$ and β:

$$\dot{\beta} = a_1\sin\psi - b_1\cos\psi + 2a_2\sin 2\psi - 2b_2\cos 2\psi \tag{3.102}$$

$$\ddot{\beta} = a_1\cos\psi + b_1\sin\psi + 4a_2\cos 2\psi + 4b_2\sin 2\psi \tag{3.103}$$

Next, Eqs. 3.86, 3.87, and 3.102 for β, $\lambda(\eta,\psi)$, and $\dot{\beta}$, respectively are substituted into Eq. 3.101. The resulting expression for $U_T^2\alpha$ is substituted into Eqs. 3.93 and 3.94. Finally, Eq. 3.86 for β and Eq. 3.103 for $\ddot{\beta}$ are substituted into Eq. 3.93.

The final versions of Eq. 3.93 and Eq. 3.94, after all of these substitutions, are very lengthy and will not be reproduced here. These equations involve products of the form $\sin n\psi \sin m\psi$, $\cos n\psi \cos m\psi$, or $\sin n\psi \cos m\psi$. These products are eliminated by the following trigonometric relations:

$$\sin(n\psi)\sin(m\psi) = \frac{1}{2}\left[\cos(n-m)\psi - \cos(n+m)\psi\right] \tag{3.104}$$

$$\cos(n\psi)\cos(m\psi) = \frac{1}{2}\left[\cos(n-m)\psi + \cos(n+m)\psi\right] \tag{3.105}$$

$$\sin(n\psi)\cos(m\psi) = \frac{1}{2}\left[\sin(n-m)\psi + \sin(n+m)\psi\right] \tag{3.106}$$

This leaves $\sin n\psi$ or $\cos n\psi$ as the only function of ψ in any individual term. To ensure that n is non-negative, use the relations:

$$\sin(-n\psi) = -\sin(n\psi) \tag{3.107}$$

$$\cos(-n\psi) = \cos(n\psi) \tag{3.108}$$

It is only necessary to retain terms for which n is less than 3 and the number of $\lambda(\eta,\psi)$ harmonics used is varied to ensure that all terms having $n < 3$ are included.

For Eq. 3.94, the azimuth integral eliminates all $\sin n\psi$ and $\cos n\psi$ terms, except for $\cos n\psi$ when $n = 0$. This leaves:

$$C_T = \frac{\sigma a}{2}\int_{\ell_1}^{1.0}\Big[\Big(\frac{\mu^2\theta_0}{2} - \frac{\mu}{2}\lambda_{1s} + \frac{\mu^2 b_2}{4} + \frac{\varepsilon\mu a_1}{2}\Big) + \Big(\frac{\mu^2\theta_1}{2} - \lambda_0 - \mu a_1 + \frac{\varepsilon\mu a_1}{2}\Big)\eta + \theta_0\eta^2 + \theta_1\eta^3\Big] \tag{3.109}$$

All of the terms not containing any harmonics of $\lambda(\eta,\psi)$ can be integrated over η:

$$C_T = \frac{\sigma a}{2}\Big[\Big(\frac{\mu^2\theta_0}{2} - \frac{\mu^2 b_2}{4} + \frac{\varepsilon\mu a_1}{2}\Big)(1-\ell_1) + \Big(\frac{\mu^2\theta_1}{2} - \mu a_1 + \frac{\varepsilon\mu a_1}{2}\Big)\Big(\frac{1-\ell_1^2}{2}\Big) + \theta_0\Big(\frac{1-\ell_1^3}{3}\Big) + \theta_1\Big(\frac{1-\ell_1^4}{4}\Big) - \int_{\ell_1}^{1.0}\Big(\frac{\mu}{2}\lambda_{1s} + \eta\lambda_0\Big)d\eta\Big] \tag{3.110}$$

This can be solved for θ_0:

$$\theta_0 = \left[\frac{6C_T}{\sigma a} - \frac{3}{4}\mu^2 b_2 (1 - l_1) - \frac{3}{4}\mu^2 \theta_1 (1 - l_1^2) - \frac{3}{4}\theta_1 (1 - l_1^4) \right]$$

$$+ \frac{3}{4}\mu a_1 (1 - l_1^2) - \frac{3}{4}\varepsilon \mu a_1 (3 - 2l_1 - l_1^2)$$

$$+ 3\int_{l_1}^{1.0} \left(\frac{\mu}{2}\lambda_{1s} + \eta \lambda_0 \right) d\eta \Bigg] \Bigg/ \left(1 - l_1^3 + \frac{3}{2}\mu^2 \right) \qquad (3.111)$$

In Eq. 3.93, there are five different types of terms: those containing no trigonometric functions of ψ and those containing either $\sin n\psi$ or $\cos n\psi$ for n = 1 or n = 2, since all terms with n > 2 have been dropped. Thus, Eq. 3.93 can be written:

$$F_0 + F_{1c}\cos\psi + F_{1s}\sin\psi + F_{2c}\cos 2\psi + F_{2s}\sin 2\psi = 0 \qquad (3.112)$$

Since Eq. 3.112 must be satisfied for all ψ between 0 and 2π, the individual factors F must each be zero. This gives five equations for the five harmonic components of β. Solving these equations:

$$a_0 = \left(\frac{a}{2\pi}\right)\left(\frac{LN}{2}\right)\left[\left(\frac{\mu^2}{2} I_0 + I_2\right)\theta_0 + \left(\frac{\mu^2}{2} I_1 + I_3\right)\theta_1 \right.$$

$$+ \left(\frac{\varepsilon}{2} I_0 + \frac{\varepsilon}{2} I_1 - I_1\right)\mu a_1 + \frac{\mu^2}{4} I_0 b_2 \qquad (3.113)$$

$$\left. - \int_{l_1}^{1.0} (\eta - \varepsilon)\left(\frac{\mu}{2}\lambda_{1s} + \eta\lambda_0\right) d\eta \right]$$

$$a_1 = \Big\{ 8\mu I_1 \theta_0 + 8\mu I_2 \theta_1 + \left[4\varepsilon\mu I_0 - 2(1+\varepsilon)\mu I_1 \right] b_2 + 4\int_{l_1}^{1.0} (\eta-\varepsilon)\left(\frac{\mu}{2}\lambda_{2c} - \mu\lambda_0 - \eta\lambda_{1s}\right) d\eta \Big\} \Big/ \left(3\mu^2 I_0 + 4I_2 - 4\varepsilon I_1\right) \tag{3.114}$$

$$b_1 = \Big\{ 4\mu I_1 a_0 + \left[2(1+\varepsilon)\mu I_1 - 4\varepsilon\mu I_0 \right] a_2 + 4\int_{l_1}^{1.0} (\eta-\varepsilon)\left(\frac{\mu}{2}\lambda_{1s} + \eta\lambda_{1c}\right) d\eta \Big\} \Big/ \left(\mu^2 I_0 + 4I_2 - 4\varepsilon I_1\right) \tag{3.115}$$

$$a_2 = \left(\frac{a}{2\pi}\right)\left(\frac{F}{3}\right)\left(\frac{LN}{2}\right) \Big\{ \frac{-\mu^2}{2} I_0 \theta_0 - \frac{\mu^2}{2} I_1 \theta_1 + \left[\left(\frac{2-\varepsilon}{2}\right)\mu I_1 - \frac{\varepsilon\mu}{2} I_0\right] a_1 + \left(2I_2 - 2\varepsilon I_1\right) b_2 + \int_{l_1}^{1.0} (\eta-\varepsilon)\left(\frac{\mu}{2}\lambda_{1s} - \frac{\mu}{2}\lambda_{3s} - \eta\lambda_{2c}\right) d\eta \Big\} \tag{3.116}$$

$$b_2 = \left(\frac{a}{2\pi}\right)\left(\frac{F}{3}\right)\left(\frac{LN}{2}\right) \Big\{ \frac{-\mu^2}{2} I_0 a_0 + \left[\left(\frac{2-\varepsilon}{2}\right)\mu I_1 - \frac{\varepsilon\mu}{2} I_0\right] b_1 + \left(2\varepsilon I_1 - 2I_2\right) a_2 - \int_{l_1}^{1.0} (\eta-\varepsilon)\left(\frac{\mu}{2}\lambda_{1c} - \frac{\mu}{2}\lambda_{3c} + \eta\lambda_{2s}\right) d\eta \tag{3.117}$$

where the integrals I are defined:

$$I_0 = \int_{\ell_1}^{1.0} (\eta - \varepsilon)\, d\eta = \frac{1}{2}\left[1 - \ell_1^2 - 2\varepsilon(1 - \ell_1)\right] \tag{3.118}$$

$$I_1 = \int_{\ell_1}^{1.0} \eta(\eta - \varepsilon)\, d\eta = \frac{1}{3}\left[1 - \ell_1^3 - \frac{3\varepsilon}{2}(1 - \ell_1^2)\right] \tag{3.119}$$

$$I_2 = \int_{\ell_1}^{1.0} \eta^2(\eta - \varepsilon)\, d\eta = \frac{1}{4}\left[1 - \ell_1^4 - \frac{4\varepsilon}{3}(1 - \ell_1^3)\right] \tag{3.120}$$

$$I_3 = \int_{\ell_1}^{1.0} \eta^3(\eta - \varepsilon)\, d\eta = \frac{1}{5}\left[1 - \ell_1^5 - \frac{5\varepsilon}{4}(1 - \ell_1^4)\right] \tag{3.121}$$

These six equations (3.111 plus 3.113-3.117) are coupled. To eliminate the a_1 - θ_0 coupling, substitute Eq. 3.114 into Eq. 3.111 and solve for θ_0:

$$\begin{aligned}\Theta_0 = \Bigg\{ & \frac{6C_T}{\sigma a} - \frac{3\mu^2}{4}(1 - \ell_1)b_2 - \frac{3\mu^2}{4}(2 - \ell_1^2 - \ell_1^4)\Theta_1 \\ & + 3\int_{\ell_1}^{1.0}\left(\frac{\mu}{2}\lambda_{1s} + \eta\lambda_0\right)d\eta + \frac{3\mu}{4}\left[\frac{(2-\varepsilon)(1-\ell_1^2) - 2\varepsilon(1-\ell_1)}{3\mu^2 I_0 + 4I_2 - 4\varepsilon I_1}\right] \\ & \left[2(2\varepsilon I_0 - \varepsilon I_1 - I_1)\mu b_2 + 8\mu\Theta_1 I_2 + \int_{\ell_1}^{1.0}(\eta - \varepsilon)\left(\frac{\mu}{2}\lambda_{2c} - \mu\lambda_0 - \eta\lambda_{1s}\right)d\eta\right]\Bigg\} \\ & \Bigg/ \left\{\left(1 + \frac{3\mu^2}{2} - \ell_1^3\right) - 6\mu^2 I_1\left[\frac{(2-\varepsilon)(1-\ell_1^2) - 2\varepsilon(1-\ell_1)}{3\mu^2 I_0 + 4I_2 - 4\varepsilon I_1}\right]\right\}\end{aligned}$$

(3.122)

To eliminate a_2 - b_2 coupoing, substitute Eq. 3.117 into Eq. 3.116 and solve for a_2:

$$
\begin{aligned}
a_2 = & \left(\frac{a}{2\pi}\right)\left(\frac{F}{3}\right)\left(\frac{LN}{2}\right)\Bigg\{\Bigg(\frac{-\mu^2}{2} I_0 \theta_0 - \frac{\mu^2}{2} I_1 \theta_1 - \frac{\mu}{2}\left[\varepsilon I_0 - (2-\varepsilon) I_1\right] a_1 \\
& + \int_{l_1}^{1.0} (\eta-\varepsilon)\left(\frac{\mu}{2}\lambda_{1s} - \frac{\mu}{2}\lambda_{3s} - \eta\lambda_{2c}\right) d\eta + \left(\frac{a}{2\pi}\right)(F)\left(\frac{LN}{3}\right)(I_2 - \varepsilon I_1) \\
& \left[\frac{-\mu^2}{2} I_0 a_0 - \frac{\mu}{2}\left(\varepsilon I_0 - 2I_1 + \varepsilon I_1\right) b_1 - \int_{l_1}^{1.0} (\eta-\varepsilon)\left(\frac{\mu}{2}\lambda_{1c} - \frac{\mu}{2}\lambda_{3c} + \eta\lambda_{2s}\right) d\eta\right]\Bigg]\Bigg\} \\
& \Bigg/ \left[1 + F^2\left(\frac{a}{2\pi}\right)^2\left(\frac{LN}{3}\right)^2 (I_2 - \varepsilon I_1)^2\right]
\end{aligned}
\tag{3.123}
$$

To save ourselves from further algebra and longer equations, note that a_2 and b_2 are normally at least an order of magnitude smaller than θ_o, a_o, a_1, and b_1. This allows the use of the following computation scheme:

1) Assume $a_2 = b_2 = 0$

2) Compute θ_o from Eq. 3.122

3) Compute a_1 from Eq. 3.114

4) Compute a_o from Eq. 3.113

5) Compute b_1 from Eq. 3.115

6) Compute a_2 from Eq. 3.123

7) Compute b_2 from Eq. 3.117

8) Repeat steps 2 through 7, using new values of a_2 and b_2. Normally, one iteration is adequate.

In the equations above all integrals over η which contain harmonics of $\lambda(\eta,\psi)$ are not performed since these harmonics are functions of η. The program LDS-73 performs these integrations numerically.

The coning angle a_o is an input to LDS-73 which must be computed by hand. A simplified expression for a_o is, therefore, developed, assuming: $\varepsilon = 0$, $F = 1.0$, $\lambda(\eta,\psi) = \lambda_o$, and that μ^3 can be neglected compared to one.

$$a_o = \left(\frac{a}{2\pi}\right)\left(\frac{LN}{4}\right)\left[\left(1 - \frac{\mu^2}{18}\right)\frac{3C_T}{\sigma a} + \left(1 - \frac{5\mu^2}{2}\right)\frac{\lambda}{12} + \left(1 + \frac{5\mu^2}{6}\right)\frac{\theta_1}{40}\right] \qquad (3.124)$$

This can be further simplified to:

$$a_o = \frac{3}{4}\left(\frac{LN}{2\pi}\right)\left(\frac{C_T}{\sigma}\right) \qquad (3.3)$$

with some sacrifice in accuracy.

It is often useful to isolate the contributions of C_T, θ_1, the various harmonics of $\lambda(\eta,\psi)$, a_2, and b_2 to θ_o, a_o, a_1, and b_1. This is done in LDS-73 by appropriate bookkeeping when this harmonic solution was programmed.

The harmonic solution given above neglects compressibility. The important compressibility effect on blade motion is the increase of the lift-curve slope $\partial C_L/\partial a$ with Mach number. An average correction factor for this (GMA) is developed in Subsection 3.3.2 (Eq. 3.161). This can be applied to the harmonic solution by replacing the lift-curve slope a with (a/GMA) wherever a appears.

3.3.2 Numerical Solution

The fundamental assumption of the numerical solution is that the blade motion consists of rigid body flapping and first mode flapwise blade bending. Given assumed mode shapes, this leads to two linearized, coupled equations for the amplitudes as a function of azimuth. These equations are solved numerically.

Since all blade motion is assumed to take place outboard of the flapping hinge, the length of the blade outboard of the flapping hinge $L = (1 - \varepsilon)R$ is often used to normalize variables instead of R. Variables normalized by L instead of R will be identified by a bar above them. Thus the normalized radial coordinate becomes:

$$\bar{\eta} = (\eta - \varepsilon)/(1 - \varepsilon) \tag{3.125}$$

The vertical displacement of point P_η on the η blade relative to the no feathering plane (NFP) is $\bar{z}(\bar{\eta},\psi)$ (normalized by L). This $\bar{z}(\bar{\eta},\psi)$ can be written in terms of two normal modes:

$$\bar{z}(\bar{\eta},\psi) = \gamma_1(\bar{\eta})\, q_1(\psi) + \gamma_2(\bar{\eta})\, q_2(\psi) \tag{3.126}$$

The first mode is rigid body flapping. Therefore:

$$\gamma_1(\bar{\eta}) = \bar{\eta} \tag{3.127}$$

$$q_1(\psi) = \beta(\psi) \tag{3.128}$$

The second mode is the first flapwise mode of blade bending. The choice of a mode shape $\gamma_2(\bar{\eta})$ is discussed in Appendix G. The mode shape normally used is:

$$\gamma_2(\bar{\eta}) = 4\bar{\eta}^2 - 3\bar{\eta} \tag{3.129}$$

based on the assumption of a uniform radial blade mass distribution $m(\bar{\eta})$ (slugs/ft) outboard of the flapping hinge.

The amplitude of the blade motion is assumed to be small enough that all terms in the blade-motion equations which are nonlinear in q_1 and q_2 can be neglected. Provision is made for a torsional spring of stiffness k (ft-lb/radian) about the flapping hinge. This spring can be used, together with flapping hinge offset, to simulate a hingeless, cantilever rotor. The resulting linearized equations of blade motion are:

$$M_5 \ddot{q}_1 + M_1 q_1 + M_2 q_2 + K\left[q_1 + q_2\left(\frac{dY_2}{d\bar{\eta}}\right)_{\bar{\eta}=0}\right] = Q_1 \tag{3.130}$$

$$\begin{aligned} &M_6 \ddot{q}_2 + M_2 q_1 + M_4 q_2 + M_7 q_2 \\ &\quad + K\left(\frac{dY_2}{d\bar{\eta}}\right)_{\bar{\eta}=0}\left[q_1 + q_2\left(\frac{dY_2}{d\bar{\eta}}\right)_{\bar{\eta}=0}\right] = Q_2 \end{aligned} \tag{3.131}$$

where the M's are various mass integrals, K is a normalized k, Q_1 and Q_2 are generalized forces due to airloads, $EI(\bar{\eta})$ is the radial distribution of flapwise stiffness (lb-ft^2) of the rotor blade, and the dot notation represents differentiation with respect to azimuth angle (ψ).

$$M_1 = \frac{2}{\rho a c R} \int_0^1 \bar{\eta}(\bar{\varepsilon} + \bar{\eta})\, m(\bar{\eta})\, d\bar{\eta} \tag{3.132}$$

$$M_2 = \frac{2}{\rho a c R} \int_0^1 \bar{\varepsilon}\; m(\bar{\eta})\, d\bar{\eta} \tag{3.133}$$

$$M_4 = \frac{2}{\rho a c R} \int_0^1 (\bar{\varepsilon} + \bar{\eta})\left[\int_0^{\bar{\eta}} \left(\frac{dY_2}{d\xi}\right)^2 d\xi\right] m(\bar{\eta})\, d\bar{\eta} \tag{3.134}$$

$$M_5 = \frac{2}{\rho a c R} \int_0^1 \bar{\eta}^2\, m(\bar{\eta})\, d\bar{\eta} \tag{3.135}$$

$$M_6 = \frac{2}{\rho a c R} \int_0^1 Y_2^2\, m(\bar{\eta})\, d\bar{\eta} \tag{3.136}$$

$$M_7 = \frac{2}{\rho a c R \Omega^2 L^4} \int_0^1 \left(\frac{d^2 Y_2}{d\bar{\eta}^2}\right)^2 EI(\bar{\eta})\, d\bar{\eta} \tag{3.137}$$

$$K = 2k \big/ \rho a c R \Omega^2 L^3 \tag{3.138}$$

$$Q_1 = (1-\varepsilon)^{-2} \int_{l_1}^{1.0} \bar{\eta} \left[\frac{2L(\eta,\psi)}{\rho a c (\Omega R)^2} \right] d\eta \tag{3.139}$$

$$Q_2 = (1-\varepsilon)^{-2} \int_{l_1}^{1} \gamma_2 \left[\frac{2L(\eta,\psi)}{\rho a c (\Omega R)^2} \right] d\eta \tag{3.140}$$

The equations for the M's can be simplified for the case of a uniform blade mass distribution [$m(\bar{\eta})$ = constant]. For typical high aspect ratio rotor blades, this implies a flapwise moment of inertia about the flapping hinge of:

$$I_b = \frac{m(\bar{\eta})\, L^3}{3} \tag{3.141}$$

From the definition of LN (Eq. 3.90),

$$LN = (1-\varepsilon)^{-3} \frac{6\pi\rho c R}{m(\bar{\eta})} \tag{3.142}$$

Assuming $\gamma_2(\bar{\eta})$ given by Eq. 3.129, the M's can now be written in terms of LN for constant $m(\bar{\eta})$:

$$M_1 = \frac{2\pi}{LNa} (1-\varepsilon)^{-3} \left[2 + \left(\frac{3\varepsilon}{1-\varepsilon} \right) \right] \tag{3.143}$$

$$M_2 = \frac{2\pi}{LNa} (1-\varepsilon)^{-3} \left[\frac{-\varepsilon}{1-\varepsilon} \right] \tag{3.144}$$

$$M_4 = \frac{2\pi}{LNa} (1-\varepsilon)^{-3} \left[\frac{38}{5} + 11 \left(\frac{\varepsilon}{1-\varepsilon} \right) \right] \tag{3.145}$$

$$M_5 = \frac{2\pi}{LNa} (1-\varepsilon)^{-3} \left[2 \right] \tag{3.146}$$

$$M_6 = \frac{2\pi}{LNa} (1-\varepsilon)^{-3} \left[\frac{6}{5} \right] \tag{3.147}$$

Assuming a uniform stiffness distribution [EI($\bar{\eta}$) = EI = constant]:

$$M_7 = \frac{128\,(1-\varepsilon)^{-4}}{\rho a c R \Omega^2 R^4}\left[EI\right] \tag{3.148}$$

The normalized k can be written in terms of R and ε instead of L:

$$K = \frac{2\,k\,(1-\varepsilon)^{-3}}{\rho a c R \Omega^2 R^3} \tag{3.149}$$

Now the equations of blade motion can be simplified for this special case:

$$M_5 \ddot{q}_1 + M_1 q_1 + M_2 q_2 + K\left[q_1 - 3q_2\right] = Q_1 \tag{3.150}$$

$$M_6 \ddot{q}_2 + M_2 q_1 + (M_4 + M_7)\, q_2 - 3K\left[q_1 - 3K\right] = Q_2 \tag{3.151}$$

These are the equations used in LDS-73, together with the $\gamma_2(\bar{\eta})$ given by Eq. 3.129.

An expression for $L(\eta,\psi)$ can be obtained by combining Eq. 3.89 and Eq. 3.94:

$$L(\eta,\psi) = \frac{1}{2}\rho a c (\Omega R)^2 \left[F\, U_T^2\, \alpha(\eta,\psi) + (1-F)\,\frac{2C_T}{\sigma a}\right] \tag{3.152}$$

The tangential airspeed U_T is computed from either Eq. 3.84, if tangential induced velocities are included, or Eq. 3.98, if they are neglected. The perpendicular airspeed U_P is computed assuming that q_1 and q_2 are small angles:

$$U_p(\eta,\psi) = \lambda(\eta,\psi) + \mu a_1 + \left[q_1 + q_2\left(\frac{d\gamma_2}{d\bar{\eta}}\right)\right]\mu \cos\psi \tag{3.153}$$

$$+\,\bar{\eta}\,\dot{q}_1 + \gamma_2(\bar{\eta})\,\dot{q}_2$$

The general expression for angle of attack is:

$$\alpha(\eta,\psi) = \theta(\eta) - \tan^{-1}\left(U_p / U_T\right) \tag{3.95}$$

For part of the numerical solution, a small angle version of this expression is used:

$$\alpha(\eta,\psi) = \theta(\eta) - U_P / U_T \tag{3.154}$$

These expressions for $\alpha(\eta,\psi)$ do not apply in the reverse flow region ($U_T < 0$). Taking only the principal value of $\tan^{-1}(U_P/U_T)$:

$$\alpha(\eta,\psi) = \begin{cases} \theta(\eta) - \tan^{-1}(U_P/U_T) & , \quad U_T > 0 \\ -\theta(\eta) + \tan^{-1}(U_P/U_T) & , \quad U_T < 0 \end{cases} \tag{3.155}$$

For the small angle version:

$$U_T^2 \, \alpha(\eta,\psi) = |U_T| \left[U_T \, \theta(\eta) - U_P \right] \tag{3.156}$$

which applies for all U_T.

Now Q_1 and Q_2 can be written using Eq. 152 and Eq. 3.156):

$$Q_1 = (1-\varepsilon)^{-2} \int_{\lambda_1}^{1} \bar{\eta} \left\{ F|U_T| \left[U_T \theta(\eta) - U_P \right] + (1-F) \frac{2C_T}{\sigma a} \right\} d\eta \tag{3.157}$$

$$Q_2 = (1-\varepsilon)^{-2} \int_{\lambda_1}^{1} Y_2 \left\{ F|U_T| \left[U_T \theta(\eta) - U_P \right] + (1-F) \frac{2C_T}{\sigma a} \right\} d\eta \tag{3.158}$$

These equations for Q_1 and Q_2 assume that the lift-curve slope is a constant. It is desirable to modify this assumption to include stall and compressibility effects. Stall will be included later. The compressibility correction is:

$$\frac{\partial C_L}{\partial \alpha} = \frac{a}{GM(\eta,\psi)} \tag{3.159}$$

$$GM(\eta,\psi) = \mathrm{Max.}\left\{ \frac{0.6}{\sqrt{1-[M(\eta,\psi)]^2}} \right\} \tag{3.160}$$

where $M(\eta,\psi)$ is the local Mach number:

$$M(\eta,\psi) = \Omega R\, U_T(\eta,\psi)\,/1100. \tag{3.161}$$

An average compressibility correction is also defined:

$$GMA = \mathrm{Max.}\left\{ \frac{0.6}{\sqrt{1-[0.75\,\Omega R/1100]^2}} \right\} \tag{3.162}$$

The average correction GMA can be applied to the lift-curve slope a used in the harmonic solution. It is not needed for the numerical solution where $GM(\eta,\psi)$ is used.

To understand the use of $GM(\eta,\psi)$, recall that a is a constant input, $\partial C_L/\partial\alpha$, and that the actual $\partial C_L/\partial\alpha$ is obtained by multiplying a by an appropriate factor. The equations of blade motion are normalized by a factor which includes a but not any factors which might multiply a. Thus, any correction factors for a multiply the angle of attack $\alpha(\eta,\psi)$ in the equations of motion, not a. This implies that:

$$Q_1 = (1-\varepsilon)^{-2}\int_{x_1}^{1} \bar{\eta}\left\{ \frac{F|U_T|}{GM(\eta,\psi)}\left[U_T\theta(\eta) - U_P\right] + (1-F)\,\frac{2C_T}{\sigma a} \right\} d\eta \tag{3.163}$$

$$Q_2 = (1-\varepsilon)^{-2}\int_{x_1}^{1} Y_2\left\{ \frac{F|U_T|}{GM(\eta,\psi)}\left[U_T\theta(\eta) - U_P\right] + (1-F)\,\frac{2C_T}{\sigma a} \right\} d\eta \tag{3.164}$$

The U_P terms in Q_1 and Q_2 contain the dependent variables q_1 and q_2. To isolate this dependence write:

$$Q_1 = C_1 - C_2 \dot{q}_1 - C_3 \dot{q}_2 - C_4 q_1 - C_5 q_2 \tag{3.165}$$

$$Q_2 = C_6 - C_3 \dot{q}_1 - C_8 \dot{q}_2 - C_9 q_1 - C_{10} q_2 \tag{3.166}$$

which implies that:

$$C_1 = (1-\varepsilon)^{-2} \int_{\ell_1}^{1} \bar{\eta} \left\{ \frac{F|U_T|}{GM(\eta,\psi)} \left[U_T \theta(\eta) - \lambda(\eta,\psi) - \mu a_1 \right] - (1-F) \frac{2C_T}{\sigma a} \right\} d\eta \tag{3.167}$$

$$C_6 = (1-\varepsilon)^{-2} \int_{\ell_1}^{1} \gamma_2 \left\{ \frac{F|U_T|}{GM(\eta,\psi)} \left[U_T \theta(\eta) - \lambda(\eta,\psi) - \mu a_1 \right] - (1-F) \frac{2C_T}{\sigma a} \right\} d\eta \tag{3.168}$$

$$C_2 = (1-\varepsilon)^{-2} \int_{\ell_1}^{1} \left[\frac{F|U_T|}{GM(\eta,\psi)} \right] \left[\bar{\eta} \right]^2 d\eta \tag{3.169}$$

$$C_3 = (1-\varepsilon)^{-2} \int_{\ell_1}^{1} \left[\frac{F|U_T|}{GM(\eta,\psi)} \right] \bar{\eta}\, \gamma_2 \, d\eta \tag{3.170}$$

$$C_8 = (1-\varepsilon)^{-2} \int_{\ell_1}^{1} \left[\frac{F|U_T|}{GM(\eta,\psi)} \right] \left[\gamma_2 \right]^2 d\eta \tag{3.171}$$

$$C_4 = (1-\varepsilon)^{-2} \int_{\ell_1}^{1} \left[\frac{F|U_T| \mu \cos\psi}{GM(\eta,\psi)} \right] \bar{\eta} \, d\eta \tag{3.172}$$

$$C_9 = (1-\varepsilon)^{-2} \int_{\ell_1}^{1} \left[\frac{F|U_T| \mu \cos\psi}{GM(\eta,\psi)} \right] \gamma_2 \, d\eta \tag{3.173}$$

$$C_5 = (1-\varepsilon)^{-2} \int_{\ell_1}^{1} \left[\frac{F|U_T| \mu \cos\psi}{GM(\eta,\psi)} \right] \bar{\eta} \left[\frac{d\gamma_2}{d\bar{\eta}} \right] d\eta \tag{3.174}$$

$$C_{10} = (1-\varepsilon)^{-2} \int_{d_1}^{1} \left[\frac{F|U_T| \mu \cos\psi}{GM(\eta,\psi)} \right] \gamma_2 \left[\frac{d\gamma_2}{d\bar{\eta}} \right] d\eta \tag{3.175}$$

For a given $\gamma_2(\bar{\eta})$, such as Eq. 129, all of the C's, except C_1 and C_6 can be integrated in closed form, if $q_T(\eta,\psi)$, the induced contribution to U_T, is neglected. This is not done in LDS-73 where the option to include $q_T(\eta,\psi)$ has been retained by using numerical integration.

Dynamic stall effects will be included in the final airloads computation (Subsection 3.4.2), but they are only very simply represented in the numerical solution for blade motion. This is because the computation of dynamic stall effects in the final airloads solution shows that they are small for the low advance ratio cases considered here. Should it become necessary, dynamic stall could be included in the numerical solution, but this would require a major revision to accurately predict $\dot{\alpha} = \partial\alpha/\partial\psi$.

For the numerical solution in LDS-73, dynamic stall is simply represented by raising the static stall angle of attack (α_s) to 20°. The C_L vs α curve used is shown in Fig. 22, where $\partial C_L/\partial\alpha$ is obtained from Eq. 3.159. To apply this $\alpha(\eta,\psi)$ is computed using Eq. 3.155 just before the C's are computed. If $|\alpha(\eta,\psi)| < \alpha_s$, then the C's are computed normally (Eq. 3.167 to Eq. 3.175). If $|\alpha(\eta,\psi)| > \alpha_s$, then all of the coefficients for q_1, q_2, $\dot{q}_1$, $\dot{q}_2$ (C_2, C_3, C_4, C_5, C_8, C_9, C_{10}) are set equal to zero, since $\partial C_L/\partial\alpha = 0$. The terms which do not depend on blade motion become:

$$C_1 = (1-\varepsilon)^{-2} \int_{d_1}^{1.0} \bar{\eta} \left[\frac{F\, U_T^2\, \alpha(\eta,\psi)}{GM(\eta,\psi)\,|\alpha(\eta,\psi)|} \alpha_s - (1-F) \frac{2C_T}{\sigma a} \right] d\eta \tag{3.176}$$

$$C_2 = (1-\varepsilon)^{-2} \int_{d_1}^{1.0} \gamma_2(\bar{\eta}) \left[\frac{F\, U_T^2\, \alpha(\eta,\psi)}{GM(\eta,\psi)\,|\alpha(\eta,\psi)|} \alpha_s - (1-F) \frac{2C_T}{\sigma a} \right] d\eta \tag{3.177}$$

Now the equations of blade motion (Eq. 3.150, 3.151) can be rewritten using these coefficients $C_1 - C_{10}$.

$$M_5 \ddot{q}_1 + C_2 \dot{q}_1 + C_3 \dot{q}_2 + (M_1 + C_4 + K) q_1 + (M_2 + C_5 - 3K) q_2 = C_1 \tag{3.178}$$

$$M_6 \ddot{q}_2 + C_3 \dot{q}_1 + C_8 \dot{q}_2 + (M_2 + C_9 - 3K) q_1 + (M_4 + M_7 + C_{10} + 9K) q_2 = C_6 \tag{3.179}$$

In review, observe that all of the blade motion (q_1, q_2) dependence in the equations of motion has been made explicit in Eqs. 178 and 3.179 with the exception of that contained in $\partial C_L/\partial\alpha$. Recall that when these equations were normalized, they were divided by a, where a is an input, constant $\partial C_L/\partial\alpha$. The actual $\partial C_L/\partial\alpha$ is a multiplied by appropriate correction factors for compressibility and stall. These lift-curve slope correction factors multiply the coefficients $C_1 - C_{10}$. The compressibility-correction factor is $[1.0/GM(\eta,\psi)]$, which is independent of blade motion. Stall is a function of angle of attack α, which is in turn a function of blade motion. Thus, the stall-correction factor for $\partial C_L/\partial\alpha$ is a function of blade motion. This dependence of $\partial C_L/\partial\alpha$ on blade motion makes the blade-motion equations nonlinear. To avoid this nonlinearity, a very simple C_L vs α curve is used (Fig. 22). Using this curve, $\partial C_L/\partial\alpha$ is either a constant, for α below stall, or $\partial C_L/\partial\alpha$ is zero, for α above stall. The choice between the two possible values of $\partial C_L/\partial\alpha$ is made at each step in the iterative, numerical blade-motion solution, by computing $\alpha(\eta,\psi)$ using the best available estimate of the blade motion.

A finite difference solution technique for equations similar to Eqs. 3.178 and 3.179 is described on pages 668 to 672 of Bisplinghoff, Ashley, and Halfman [56]. The basis of this technique is a set of formulas for the first and second derivatives of blade motion $q(\psi)$ with respect to azimuth angle ψ in terms of the blade motion at the last four azimuth angle steps:

$$\dot{q}(\psi) = \frac{11q(\psi) - 18q(\psi-\Delta\psi) + 9q(\psi-2\Delta\psi) - 2q(\psi-3\Delta\psi)}{6\Delta\psi} \tag{3.180}$$

$$\ddot{q}(\psi) = \frac{2q(\psi) - 5q(\psi-\Delta\psi) + 4q(\psi-2\Delta\psi) - q(\psi-3\Delta\psi)}{(\Delta\psi)^2} \tag{3.181}$$

These formulas can be used to compute $\dot{q}_1(\psi)$, $\dot{q}_2(\psi)$, $\ddot{q}_1(\psi)$, and $\ddot{q}_2(\psi)$ and the resulting expressions can be substituted into Eqs. 3.178 and 3.179. This leads to the following equations for $q_1(\psi)$ and $q_2(\psi)$:

$$a_{11}\, q_1(\psi) + a_{12}\, q_2(\psi) = y_1 \tag{3.182}$$

$$a_{21}\, q_1(\psi) + a_{22}\, q_2(\psi) = y_2 \tag{3.183}$$

where:

$$a_{11} = \frac{2M_5}{(\Delta\psi)^2} + \frac{11C_2}{6\Delta\psi} + M_1 + K + C_4 \tag{3.184}$$

$$a_{12} = \frac{11C_3}{6\Delta\psi} + M_2 + C_5 - 3K \tag{3.185}$$

$$a_{21} = \frac{11C_3}{6\Delta\psi} + M_2 + C_9 - 3K \tag{3.186}$$

$$a_{22} = \frac{2M_6}{(\Delta\psi)^2} + \frac{11C_8}{6\Delta\psi} + M_4 + M_7 + C_{10} + 9K \tag{3.187}$$

$$\begin{aligned} y_1 = C_1 &+ \left[\frac{5M_5}{(\Delta\psi)^2} + \frac{3C_2}{\Delta\psi}\right] q_1(\psi-\Delta\psi) - \left[\frac{4M_5}{(\Delta\psi)^2} + \frac{3C_2}{2\Delta\psi}\right] q_1(\psi-2\Delta\psi) \\ &+ \left[\frac{M_5}{(\Delta\psi)^2} + \frac{C_2}{3\Delta\psi}\right] q_1(\psi-3\Delta\psi) + \frac{3C_3}{\Delta\psi} q_2(\psi-\Delta\psi) \\ &- \frac{3C_3}{2\Delta\psi} q_2(\psi-2\Delta\psi) + \frac{C_3}{3\Delta\psi} q_2(\psi-3\Delta\psi) \end{aligned} \tag{3.188}$$

$$\begin{aligned} y_2 = C_6 &+ \left[\frac{5M_6}{(\Delta\psi)^2} + \frac{3C_8}{\Delta\psi}\right] q_2(\psi-\Delta\psi) - \left[\frac{4M_6}{(\Delta\psi)^2} + \frac{3C_8}{2\Delta\psi}\right] q_2(\psi-2\Delta\psi) \\ &+ \left[\frac{M_6}{(\Delta\psi)^2} + \frac{C_8}{3\Delta\psi}\right] q_2(\psi-3\Delta\psi) + \frac{3C_3}{\Delta\psi} q_1(\psi-\Delta\psi) \\ &- \frac{3C_3}{2\Delta\psi} q_1(\psi-2\Delta\psi) + \frac{C_3}{3\Delta\psi} q_1(\psi-3\Delta\psi) \end{aligned} \tag{2.189}$$

Solving these equations for $q_1(\psi)$ and $q_2(\psi)$:

$$q_1(\psi) = \frac{a_{22} y_1 - a_{12} y_2}{a_{11} a_{22} - a_{12} a_{21}} \tag{3.190}$$

$$q_2(\psi) = \frac{a_{11} y_2 - a_{21} y_1}{a_{11} a_{22} - a_{12} a_{21}} \tag{3.191}$$

The results of the harmonic solution for blade motion are used to initialize the numerical solution. Thus the initial values for blade motion are:

$$q_1(\psi) = a_0 - a_1 \cos\psi - b_1 \sin\psi - a_2 \cos 2\psi - b_2 \sin 2\psi \tag{3.192}$$

$$\dot{q}_1(\psi) = a_1 \sin\psi - b_1 \cos\psi + 2a_2 \sin 2\psi - 2b_2 \cos 2\psi \tag{3.193}$$

$$q_2(\psi) = 0.0 \tag{3.194}$$

$$\dot{q}_2(\psi) = 0.0 \tag{3.195}$$

The harmonic solution result for θ_o is also used as an initial value for the numerical solution. The initial azimuthal step size $\Delta\psi$ used in the numerical solution is the same as the $\Delta\psi$ used elsewhere in LDS-73. The solution proceeds as follows:

1) Initialize the blade pitch using θ_o from the harmonic solution and an input twist $\theta_T(\eta)$

$$\theta(\eta) = \theta_0 + \theta_T(\eta) \tag{3.196}$$

2) Initialize q_1, $\dot{q}_1$, q_2, $\dot{q}_2$ for $\psi = 0, \Delta\psi, 2\Delta\psi, \ldots 2\pi$ using Eqs. 3.192 through 3.195.

3) Start loop where ψ runs from 0 to 2π in steps of $\Delta\psi$

4) Compute $\alpha(\eta,\psi)$ using Eq. 3.155.

5) Compute $C_1 - C_{10}$ using either Eqs. 3.167 through 3.175 or Eqs. 3.176 through 3.177 depending on the value of $\alpha(\eta,\psi)$

6) Compute new values for $q_1(\psi)$ and $q_2(\psi)$ using Eqs. 3.190 through 3.191.

7) Compute new values of $\dot{q}_1(\psi)$ and $\dot{q}_2(\psi)$ using $q_1(\psi)$ and $q_2(\psi)$ from step (6) in Eq. 3.180.

8) End of ψ loop: either increment ψ by $\Delta\psi$ and go to step (4)or, if $\psi = 2\pi$, go to step (9)

9) Check for convergence. Require:

$$|q_1(0) - q_1(2\pi)| < EQ \tag{3.197}$$

$$|q_2(0) - q_2(2\pi)| < EQ \tag{3.198}$$

$$|\dot{q}_1(0) - \dot{q}_1(2\pi)| < EQ \tag{3.199}$$

$$|\dot{q}_2(0) - \dot{q}_2(2\pi)| < EQ \tag{3.200}$$

for convergence, where EQ is an input (normally $EQ = 10^{-5}$). If converged,go to step (12), otherwise go to step (10).

10) If more than NQIM (input, normally NQIM = 25), iterations have been done, halve $\Delta\psi$ and go back to step (2). Since $\Delta\psi$ is now less than the value used by the rest of LDS-73, linear interpolation

is used for $\lambda(\eta,\psi)$. Halving $\Delta\psi$ is only attempted NQCM (input, normally NQCM = 1) times.

11) If less than NQIM iterations have been done, harmonically analyze $q_1(\psi)$ to obtain a_1 which updates the value of μa_1 used in computing $\alpha(\eta,\psi)$. Set q_1, q_2, $\dot{q}_1$, $\dot{q}_2$ equal to the weighted average of their values from the last two iterations (WF = input weighting factor, normally WF = 0.5). Set: $q_1(o) = q_1(2\pi)$, $q_2(o) = q_2(2\pi)$, $\dot{q}_1(o) = \dot{q}_1(2\pi)$, and $\dot{q}_2(o) = \dot{q}_2(2\pi)$ and go to step (3).

12) Compute C_T using $\alpha(\eta,\psi)$ from step (5) in Eq. 3.94 (modified to include stall and compressibility effects)

13) Check for C_T convergence. Require

$$|C_T - C_{T_o}| < ECT \tag{3.201}$$

where C_{T_o} is the required C_T(input) and ECT is an input (normally ECT = 2×10^{-5}). If converged, the solution is complete. Otherwise change θ_o:

$$\theta_o = \theta_o + (C_{T_o} - C_T)\frac{\partial\theta_o}{\partial C_T} \tag{3.202}$$

where $\partial\theta_o/\partial C_T$ is readily obtained by differentiating Eq. 3.122 with respect to C_T. Initialize $\theta(\eta)$ using Eq. 3.196 and go to step (3).

3.4 Airloads and Performance

This subsection assumes that the downwash $\lambda(\eta,\psi)$ and the blade motion are known. The task here is to compute rotor airloads and performance. The aerodynamic forces can be resolved in various ways. The aerodynamic forces computed here are: airloads (lift), drag, thrust, horizontal force, azimuthal force, and radial force. All of these are computed at each point P_η on the rotor and are output either in (lb/in) or, for the drag, in coefficient form. The total thrust, horizontal force and propulsive force for the entire rotor are computed as output in coefficient form. Torque coefficient is computed including a breakdown into parasite torque and induced torque. Finally, rotor equivalent lift to drag ratio (L/D) and total vehicle L/D are computed.

The angle of attack $\alpha(\eta,\psi)$ is easily computed when downwash and blade motion are known. The important part of the computations in this subsection is therefore the C_L vs α and C_D vs α curves for a two-dimensional airfoil section. These curves are functions of Mach number and rate of change of angle of attack ($\dot{\alpha} = \partial\alpha/\partial\psi$). For a production airloads computation, these airfoil section characteristics would normally be input data tables for each particular section used. For LDS-73, this is not done. Instead, a set of simplified functions is used to represent a generalized, cambered helicopter airfoil section. The representation used here is much more sophisticated than that used in the numerical blade motion solution, but it is not complicated enough to show the effects of different airfoil sections on rotor performance. It is anticipated that an industrial user of this work would modify this subsection to reflect his own airfoil section data.

3.4.1 Angle of Attack

The first step is to compute the angle of attack $\alpha(\eta,\psi)$ and its time rate of change $\dot{\alpha}(\eta,\psi) = \partial\alpha/\partial\psi$. The angle of attack $\alpha(\eta,\psi)$ is computed using Eq. 3.155, repeated here:

$$\alpha(\eta,\psi) = \begin{cases} \theta(\eta) - \tan^{-1}(U_P/U_T) \ , & U_T > 0 \\ -\theta(\eta) + \tan^{-1}(U_P/U_T) \ , & U_T < 0 \end{cases} \tag{3.155}$$

The tangential airspeed U_T is computed using either Eq. 3.84, if tangential induced velocities are included, or Eq. 3.98, if they are neglected. The equation used for the perpendicular airspeed U_P depends on which blade motion solution was used. For the harmonic solution, Eq. 3.99 is used, where β and $\dot{\beta}$ are obtained from Eq. 3.102 and Eq. 3.103, respectively. For the numerical solution, Eq. 3.152 is used.

The time rate of change of angle of attack is approximated by:

$$\dot{\alpha}(\eta,\psi) = \left[\alpha(\eta,\psi) - \alpha(\eta,\psi-\Delta\psi)\right] / \Delta\psi \tag{3.203}$$

3.4.2 Airfoil Section Characteristics - Lift

The technique used here to represent radial flow and dynamic stall effects on lift is adapted from Harris [57]. This technique represents these effects by the use of an artificial $\partial C_L/\partial\alpha$. The original technique [57] is adaptable to any airfoil section, but the specialized version used here is intended to represent a typical, cambered helicopter airfoil section.

Define the chordwise and spanwise (radial) components of the total velocity in the TPP at point P_η. The chordwise component is:

$$V_{CH} = U_T(\eta,\psi) \tag{3.204}$$

where U_T comes from Eq. 3.84 or Eq. 3.98. The spanwise (radial) component is:

$$V_{SP} = \mu \cos\psi \tag{3.205}$$

The total (resultant) velocity is:

$$V_R = \sqrt{V_{CH}^2 + V_{SP}^2} \tag{3.206}$$

To avoid $|\alpha(\eta,\psi)| > \pi/2$ define α' such that $|\alpha'| \leq \pi/2$:

$$\alpha' = \begin{cases} \alpha(\eta,\psi) & , \alpha_m \leq \pi/2 \\ -\dfrac{\alpha(\eta,\psi)}{\alpha_m}\left[\pi - \alpha_m\right] & , \alpha_m > \pi/2 \end{cases} \tag{3.207}$$

where α_m is:

$$\alpha_m = |\alpha(\eta,\psi)| \tag{3.208}$$

If $\alpha_m > \pi/2$, it is also necessary to multiply V_{CH} by minus one (-1) to make V_{CH} compatible with the definition of α'.

The artificial $\partial C_L/\partial\alpha$ is determined using a reference angle of attack α_R. This α_R is obtained by subtracting a dynamic stall $\Delta\alpha$ from α'. Define:

$$\Delta\alpha_{DS} = 1.076 \ln\left(\frac{0.6}{M_B}\right)\sqrt{b\dot{\alpha}/RV_{CH}} \tag{3.209}$$

where b is the blade semichord and M_B is the chordwise Mach number at point P_η modified so that:

$$0.2 \leq M_B \leq 0.6 \tag{3.210}$$

To prevent dynamic stall effects from changing the sign of the lift, we require:

$$\Delta\alpha_{DS} \leq \alpha_m \tag{3.211}$$

Now:

$$\alpha_R = \left| \left[\alpha' - \Delta\alpha_{DS} \left(\frac{\dot{\alpha}}{|\dot{\alpha}|} \right) \right] \right| \tag{2.212}$$

Further, we require that:

$$\frac{\alpha_m}{2} \leq \alpha_R \leq \frac{\pi}{2} \tag{3.213}$$

The static stall angle of attack (α_{ss}) and lift coefficient $C_{L_{ss}}$ are needed. For the typical cambered helicopter airfoil section considered here:

$$\alpha_{ss} = 0.26 - \left[\frac{0.0065}{0.65 - M_B} \right] \tag{3.214}$$

except that in reverse flow ($V_{CH} < 0$) or for a negative angle of attack ($\alpha' < 0$):

$$\alpha_{ss} = 0.13 \tag{3.215}$$

The static stall C_L is:

$$C_{L_{ss}} = \alpha_{ss} \left[\frac{a}{GM(\eta,\psi)} \right] \tag{3.216}$$

where $GM(\eta,\psi)$ is the compressibility factor, Eq. 3.160).

Now the artificial $\partial C_L/\partial\alpha$ (labeled a_R) can be computed. There are several cases. If $\alpha_R \leq \alpha_{ss}$:

$$a_R = a/GM(\eta,\psi) \tag{3.217}$$

If $\alpha_R > \alpha_{ss}$:

$$a_R = \text{Max.} \begin{Bmatrix} \left[a/GM(\eta,\psi) \right] \\ \left| \left(\dfrac{C_{L_s}}{\alpha_R} \right) \left(\dfrac{V_R}{V_{CH}} \right) \right| \end{Bmatrix} \tag{3.218}$$

where the factor (V_R/V_{CH}) accounts for radial flow and C_{L_s} is:

$$C_{L_s} = \begin{cases} C_{L_{ss}} + \left[\dfrac{(1.0 - C_{L_{ss}})(\alpha_R - \alpha_{ss})}{0.28 - \alpha_{ss}} \right] & , (0 < \alpha_R < 0.28) \\ 1.0 & , (0.28 < \alpha_R < \pi/4) \\ \sin \alpha_R & , (\pi/4 < \alpha_R < \pi/2) \end{cases} \tag{3.219}$$

Finally, the lift coefficient C_L, including radial flow and dynamic stall effects, is:

$$C_L = a_R \alpha' \tag{3.220}$$

where α' is defined by Eq. 3.207. To illustrate the results obtained using these procedures, Fig. 23 shows the magnitude of C_L vs the magnitude of α (degrees) for the static case ($\dot{\alpha} = 0$) at Mach numbers of 0.2, 0.5, and 0.9.

3.4.3 Airfoil Section Characteristics - Drag

The effect of dynamic stall on drag is represented by using the reference angle of attack α_R, computed in Subsection 3.4.2 in place of the actual angle of attack. The static stall angle of attack α_{ss} is used as computed in Subsection 3.4.2 with one exception: for negative angle of attack ($\alpha' < 0$):

$$\alpha_{ss} = 0.07 \tag{3.221}$$

The incompressible drag coefficient C_D is computed first and a compressibility correction C_{D_m} is added later.

The computation of the incompressible C_D involves four formulas: one for the unstalled case plus three stalled cases. Two parameters used in these formulas are: C_{D_o} (C_D at zero α_R) and $C_{D_{ss}}$ (C_D at $\alpha_R = \alpha_{ss}$):

$$C_{D_o} = 0.0087 \left| V_R / V_{CH} \right| \tag{3.222}$$

where the factor $|V_R/V_{CH}|$ corrects C_{D_o} for radial flow.

$$C_{D_{ss}} = C_{D_o} - .0216\,\alpha_{ss} + .40\,(\alpha_{ss})^2 \tag{3.223}$$

Now the incompressible C_D is:

$$C_D = \begin{cases} C_{D_o} - .0216\,\alpha_R + .40\,(\alpha_R)^2 & , \; (0 < \alpha_R < \alpha_{ss}) \\ C_{D_{ss}} + (.2 - C_{D_{ss}}) \left(\dfrac{\alpha_R - \alpha_{ss}}{.26 - \alpha_{ss}} \right) & , \; (\alpha_{ss} < \alpha_R < .26) \\ .20 + 1.8\,(\alpha_R - .26) & , \; (.26 < \alpha_R < 1.26) \\ 2.0 & , \; (1.26 < \alpha_R) \end{cases} \tag{3.224}$$

The compressibility correction C_{D_m} is based on the drag divergence Mach number M_D.

$$M_D = 0.85 - 0.33 \left| C_L \right| \tag{3.225}$$

where C_L is obtained from Eq. 3.220. If the incompressible C_D is greater than 0.018, then modify M_D:

$$M_D \rightarrow M_D - \sqrt{C_D - 0.018} \tag{3.226}$$

Also require that:

$$M_D > 0.10 \tag{3.227}$$

Now C_{D_m} is:

$$C_{D_m} = \begin{cases} 0.0 & , [M < M_D] \\ (M - M_D)^2 & , [0 < (M - M_D < .15] \\ .0225 + .4(M - M_D - .15) & , [(M - M_D) > .15] \end{cases} \tag{3.228}$$

Finally, C_{D_m} is added to C_D:

$$C_D \rightarrow C_D + C_{D_m} \tag{3.229}$$

To illustrate the results obtained using these formulas C_D vs α_R (degrees) is shown in Fig. 24 for Mach numbers of: 0.2, 0.5, and 0.9. Note that α_R is in radians in the formulas quoted here.

3.4.4 Aerodynamic Forces and Performance

The computation of aerodynamic forces, given C_L and C_D, is mostly a matter of resolving these forces in various directions. The local aerodynamic forces computed are: airloads (lift) perpendicular to the local relative wind, thrust perpendicular to the tip path plane (TPP), azimuthal force perpendicular to the blade radial axis and parallel to the TPP, the radial force parallel to the blade axis and parallel to the TPP, and the horizontal force in the plane of the direction of flight and parallel to the TPP. The total aerodynamic forces on the rotor computed here are: thrust perpendicular to the TPP, horizontal force in the plane of the direction of flight but parallel to the TPP, and propulsive force parallel to the direction of flight. The local parasite and induced torque coefficient increments as well as the total rotor torque coefficients are computed. Finally, rotor and total aircraft equivalent lift to drag ratios are computed.

Various angles are needed to resolve these forces. The angle between the local relative wind and the TPP is:

$$\alpha_{TPP} = \alpha(\eta, \psi) - \theta(\eta) - A_1 \cos\psi - B_1 \sin\psi \tag{3.230}$$

where A_1 and B_1 are cyclic pitch components relative to the TPP. Recalling the relationship between cyclic pitch and first harmonic flapping (a_1 and b_1) in Gessow-Myers [58]:

$$\alpha_{TPP} = \alpha(\eta, \psi) - \theta(\eta) + b_1 \cos\psi - a_1 \sin\psi \tag{3.231}$$

The angle between the blade element at point P_η and the TPP in the radial direction is assumed to be small. Therefore it is:

$$\frac{\partial z(\bar{\eta}, \psi)}{\partial \bar{\eta}} = q_1 + \left(\frac{\partial \gamma_2}{\partial \bar{\eta}}\right) q_2 - q_{1c} \cos\psi - q_{1s} \sin\psi \tag{3.232}$$

where $q_{1c} = - a_1$ and $q_{1s} = - b_1$ are the first harmonic sine and cosine components of $q_1(\psi)$.

The local airloading (lift) at point P_η is:

$$L(\eta, \psi) = \frac{1}{2} \rho c (\Omega R)^2 (V_{CH})^2 C_L \tag{3.233}$$

which includes dynamic stall effects in the computation of C_L (Subsection 3.4.2). The program LDS-73 also computes another $L(\eta, \psi)$ without dynamic stall for comparison with that obtained using Eq. 3.233. This simplified $L(\eta, \psi)$ is based on $\alpha(\eta, \psi)$ computed from Eq. 3.154 and a modified version of the C_L vs α curve of Fig. 22. The modification to Fig. 22 is to require that C_L go linearly to zero at $\alpha = \pm \pi/2$ with the decay starting at $\alpha \pm (\pi/2 - \alpha_s)$. Thus the positive- and negative-α portions of the C_L vs α curve become symmetrical about $\alpha = \pm \pi/4$.

The other aerodynamic forces include components of drag as well as lift. Define the azimuth drag D_A and the radial drag D_R:

$$D_A = \frac{1}{2}\rho c(\Omega R)^2 (V_{CH})^2 C_D \tag{3.234}$$

$$D_R = \frac{1}{2}\rho c(\Omega R)^2 V_{SP} V_R (.0087) \tag{3.235}$$

where V_{SP} and V_R are defined by Eq. 3.205 and Eq. 3.206, respectively. Both the lift $L(\eta,\psi)$ and the drags D_A and D_R will have the dimensions of the factor $1/2\ \rho c(\Omega R)^2$. Using the normal dimensions for ρ, c, Ω, and R gives (lb/ft). Since experimental data is normally quoted in (lb/in) the various local forces are output by LDS-73 in (lb/in). Therefore $L(\eta,\psi)$, D_A, and D_R will be understood to have dimensions of (lb/in), which involves either measuring the chord (c) in inches instead of feet or replacing the 1/2 by 1/24 in Eq. 3.233 through Eq. 3.235.

The local thrust (lb/in) can now be computed:

$$T_F(\eta,\psi) = L(\eta,\psi)\cos\alpha_{TPP} + D_A \sin\alpha_{TPP} \tag{3.236}$$

The local azimuthal force (lb/in) is:

$$A_F(\eta,\psi) = D_A\cos\alpha_{TPP} - L(\eta,\psi)\sin\alpha_{TPP} \tag{3.237}$$

The local radial force (lb/in) is:

$$R_F(\eta,\psi) = D_R - L(\eta,\psi)\sin\left[\frac{\partial z(\bar{\eta},\psi)}{\partial\bar{\eta}}\right] \tag{3.238}$$

The local horizontal force (lb/in) is:

$$H_F(\eta,\psi) = R_F(\eta,\psi)\cos\psi + A_F(\eta,\psi)\sin\psi \tag{3.239}$$

The local thrust $T_F(\eta,\psi)$ and local horizontal force $H_F(\eta,\psi)$ are integrated over the rotor disk to obtain the total thrust (lb) and the total horizontal force (lb).

$$T = \frac{R'}{2\pi} \int_0^{2\pi} \int_0^1 T_F(\eta, \psi)\, d\eta\, d\psi \tag{3.240}$$

$$H = \frac{R'}{2\pi} \int_0^{2\pi} \int_0^1 H_F(\eta, \psi)\, d\eta\, d\psi \tag{3.241}$$

where R' is R measured in inches to make the units consistent. The total propulsive force X(lb) is:

$$X = T \sin i - H \cos i \tag{3.242}$$

where i is the TPP incidence which was originally estimated (Eq. 3.4) by assuming that H is negligible compared to T.

The local increments to the total parasite torque, induced torque, and thrust coefficients are computed to show the contribution of each point P_η to the overall rotor performance.

These local increments represent the contribution of a local area, having dimensions $\Delta\eta$ by $\Delta\psi$, around point P_η, to the total coefficients for all n_b blades. The thrust coefficient increment is:

$$\Delta C_T(\eta, \psi) = \frac{n_b\, \Delta\eta\, R'}{\rho \pi R^2 (\Omega R)^2}\; \frac{\Delta\psi}{2\pi}\; T_F(\eta, \psi) \tag{3.243}$$

where R' is R measured in inches to cancel the inches in T_F (lb/in). The parasite torque coefficient increment is:

$$\Delta C_{Q_p}(\eta, \psi) = \frac{n_b\, \eta\, \Delta\eta\, R'}{\rho \pi R^2 (\Omega R)^2} \left(\frac{\Delta\psi}{2\pi}\right) D_A \cos \alpha_{TPP} \tag{3.244}$$

The induced torque coefficient increment is:

$$\Delta C_{Q_i}(\eta, \psi) = \frac{n_b\, \eta\, \Delta\eta\, R'}{\rho \pi R^2 (\Omega R)^2} \left(\frac{\Delta\psi}{2\pi}\right) \left[-L(\eta, \psi) \sin \alpha_{TPP}\right] \tag{3.245}$$

The total C_T, C_{Q_p}, and C_{Q_i} are obtained by summing these increments over the rotor disc. The total torque coefficient is the sum of the parasite and induced contributions.

$$C_Q = C_{Q_p} + C_{Q_i} \tag{3.246}$$

The overall aircraft lift to equivalent drag ratio is a useful measure of aerodynamic efficiency.

$$L/D_E = \mu C_T / C_Q \tag{3.247}$$

The rotor lift to equivalent drag ratio removes the fuselage and hub drag giving a measure of the aerodynamic efficiency of the rotor in isolation.

$$(L/D_E)_R = \mu C_T / (C_Q - \mu C_x) \tag{3.248}$$

where C_X is the propulsive force coefficient:

$$C_x = \frac{X}{\rho \pi R^2 (\Omega R)^2} \tag{3.249}$$

Finally, an angle of attack distribution $\alpha(\eta,\psi)$ is required for use in the computation of the bound circulation distribution, either for another iteration of LDS-73 or as input to a tip vortex geometry computation by WG-71. Since all circulation computations use the input, constant lift curve slope a the $\alpha(\eta,\psi)$ is adjusted to give the correct lift when $\partial C_L/\partial\alpha = a$.

$$\alpha(\eta,\psi) = C_L / a \tag{3.250}$$

where C_L is obtained from Eq. 3.220.

This completes the computation of airloads.

SECTION 4

ROTOR TIP VORTEX GEOMETRY COMPUTATION

The concentrated tip vortex line is the dominant part of the rotor wake (see Subsection 1.1). It is, therefore, worth expending extra effort to obtain a more accurate estimate for the tip vortex-line geometry than the basic rigid-wake assumption (Subsection 1.5.1). The more accurate tip-vortex geometry is called the distorted tip-vortex geometry, compared to the normal rigid-wake-geometry assumption. The computation of this distorted geometry consists of the computation of the tip-vortex distortion $\vec{D}(\psi,\delta)$. This $\vec{D}(\psi,\delta)$ is the displacement of point P_η on the tip vortex due to the induced velocity (see Subsection 1.5).

A computer program (WG-71) has been developed to compute the tip-vortex distortion $\vec{D}(\psi,\delta)$. This program is designed to be used together with the airloads program LDS-73. The wake geometry program WG-71 uses a circulation distribution which is computed by LDS-73 and which remains constant throughout the computation of $\vec{D}(\psi,\delta)$. The relationship between the airloads computation (LDS-73) and the tip-vortex-geometry computation (WG-71) is shown in Fig. 3.

The computation of the tip-vortex distortion $\vec{D}(\psi,\delta)$ is conceptually very simple. The induced velocity at various points P_η on the tip vortex is computed as a function of age (time) δ. This induced velocity is integrated over time to obtain $\vec{D}(\psi,\delta)$. The problem is that the induced velocity is a function of the tip vortex geometry which is,in turn, a function of $\vec{D}(\psi,\delta)$. Thus, as the computation proceeds and the current estimate of $\vec{D}(\psi,\delta)$ changes, it is necessary to recompute, or update, the induced velocity. This repeated updating of the induced velocity is very expensive, which means that a simple, brute-force tip-vortex-geometry computation is very expensive. It is, therefore, important to develop various refinements to the tip-vortex-geometry computation which reduce the expense of updating. These refinements greatly reduce the expense of the computation by reducing the amount of repetition required. They also greatly complicate what is basically a simple problem.

The distortion computation will be described in two stages. First, the fundamental computation is described to give a clear understanding of the basic approach used. Then the various refinements are described. These refinements complicate the computation, but they reduce the expense of the computation by reducing the amount of updating required.

4.1 Fundamental Distortion Computation

This subsection describes the fundamental distortion computation without the complication of the various refinements designed to reduce the amount of updating required. When the fundamental distortion computation is understood, the various refinements will be considered in Subsection 4.2. Since it is very important to save computational expense by the use of the simplest wake model which can give adequate accuracy, the choice of wake models will also be reviewed.

4.1.1 Basic Notation

A review of Subsection 1.5 will clarify the nomenclature and notation used. It is important to understand the differences between the notation used for airloads computations and that used for tip vortex distortion computations. The general point P_η, at which the induced velocity is computed, is located on the η blade for airloads computations, but it is located on the tip vortex for distortion computations. For airloads computations, the age of point P_η is always zero while the age of point P_ℓ is labeled δ. For distortion computations, the age of point P_η can be nonzero and is labeled δ while the age of point P_ℓ becomes δ_ℓ. This change in the definition of δ must be remembered.

The points P_η and P_ℓ are identified by the parameters (η,ψ,δ) and (ℓ,ϕ,δ_ℓ), respectively. These parameters are measured with respect to the fundamental skewed helix not the actual distorted tip vortex. Thus, these parameters remain independent of changes in the distortion. For example: η and ℓ are always equal to unity for the tip vortex. To obtain the location of point P_η or point P_ℓ, the corresponding point on the fundamental skewed helix (P_η^F or P_ℓ^F) is found using these parameters. Then the appropriate distortion, $\vec{D}(\psi,\delta)$ or $\vec{D}(\phi,\delta_\ell)$, provides a vector linking point P_η^F or point P_ℓ^F with the actual point P_η or point P_ℓ on the distorted tip vortex.

The function $\vec{D}(\psi,\delta)$ is only evaluated for a discrete set of ψ and δ.

$$\psi = 0, \Delta\psi, 2\Delta\psi, \cdots, 2\pi \tag{4.1}$$

$$\delta = 0, \Delta\psi, 2\Delta\psi, \cdots, 2\pi m \tag{4.2}$$

The $\Delta\psi$ used for a distortion computation is chosen to be the same as the $\Delta\phi$ used for the corresponding airloads computation. This is done to ensure that the distortion is computed at the end points of the tip-vortex line segments used in the airloads computation (i.e., at points P_ℓ for the airloads computation). The azimuth angle ψ only varies over the range 0 to 2π because the wake geometry is assumed to be cyclic for the steady-state-flight cases considered here. In other words if photographs of the rotor and its wake are taken at two different times, chosen such that the azimuth angles of the rotor blades are identical in both photographs, then the wake geometry will also be identical.

The azimuth angle ψ can be thought of as identifying the general location of point P_η in the wake. The age δ locates point P_η relative to the current position of the rotor. Since only the wake near the rotor is of interest for airloads computations, the tip-vortex distortion is only computed for δ ranging from 0 to $2\pi m$. A typical value of m is two.

The distortion $\vec{D}(\phi,\delta_\ell)$ is the same function as $\vec{D}(\psi,\delta)$, but is evaluated at point P_ℓ instead of at point P_η. Therefore, ϕ and δ_ℓ must also come in increments of $\Delta\psi$ to be compatible with ψ and δ. The range of δ_ℓ will be discussed in Subsection 4.1.3. The desired range of ϕ is 0 to 2π. However, solving Eq. 1.2 for ϕ yields:

$$\phi = \psi + \zeta + \delta - \delta_\ell \tag{4.3}$$

Examination of this equation for ψ and ζ between 0 and 2π and for δ and δ_ℓ between 0 and $2\pi m$ shows that ϕ is not restricted to the range 0 to 2π. Thus, it must be understood that when evaluating functions which are cyclic in azimuth angle, such as $\vec{D}(\phi,\delta_\ell)$ and $\gamma_{bv}(\phi)$, at azimuth angle ϕ, the principle value of ϕ is used (i.e., the appropriate integer multiple of 2π is added or subtracted so that: $0 \leq \phi \leq 2\pi$).

The same x,y,z coordinate system with origin at point P_η is used for distortion computations as was used for airloads computations (Subsection 3.2.3). The definitions of $\vec{a}$ and $\vec{b}$ (Eqs. 3.9 and 3.10) remain the same as well. The components of $\vec{a}$ and $\vec{b}$ are different, however, because point P_η is now in the wake instead of on the η blade.

4.1.2 Wake Models and Circulation

The wake models used are generally described in Section 2. There are some differences between the wake models used by WG-71 and those used by LDS-73. These differences are primarily because the airloads program LDS-73 was rewritten more recently than the wake geometry program WG-71. Thus, WG-71 does not have provision for the tip vortex sheet plus vortex line (VSVL) model described in Subsection 2.5.3 or for vortex sheets having a vortex core. However, WG-71 retains provisions for an n trailing vortex line model (Subsection 2.2), which was discarded by LDS-73 to reduce the size of the QMAT matrix. Finally, the trailing and shed vortex sheet models used by WG-71 are somewhat more complicated than those used by LDS-73.

The fundamental difference between the WG-71 and LDS-73 inboard trailing vortex sheet models is in the determination of $\ell_i(\phi)$, the outboard edge of the vortex sheet. For LDS-73, $\ell_i(\phi)$ is computed by requiring that the area under the curve $\gamma(\eta,\phi)$ vs. η be the same for the model bound circulation distribution as for the actual bound circulation distribution.

$$\gamma_{bv}(\phi)\left[1.0 - 0.5\,\ell_i(\phi) - 0.5\,\ell_1\right] = \int_{\ell_1}^{1.0} \gamma(\eta,\phi)\, d\eta \tag{4.4}$$

where $\gamma_{bv}(\phi)$ is defined in Subsection 2.8.1. The $\ell_i(\phi)$ obtained from Eq. 4.6 is then averaged over ϕ to obtain ℓ_2. If this ℓ_2 is greater than an input ℓ_2, the input value is used. This technique was at one time used for airloads computations and the actual $\ell_i(\phi)$ was used instead of the average value ℓ_2. This lead to difficulties with circulation iteration convergence and was abandoned in favor of the average ℓ_2. This, in turn, was replaced by an input $\ell_i(\phi)$ in LDS-73; however, WG-71 has not been updated to include this option.

WG-71 has provision for either a one segment or two segment shed vortex-sheet model, see Subsection 2.8.2. The two-segment model is used with the inboard trailing vortex-sheet model. When a shed vortex-sheet model is used, the trailing vortex sheet or line segments have a linear circulation variation to conserve circulation.

Since point P_η is not on a rotor blade in distortion computations, the near wake effects associated with the rotor blade do not apply. However, the computation of the induced velocity contributions of the tip-vortex-line and shed-vortex-sheet segments adjacent to point P_η must be done with special care. The use of the normal straight tip vortex-line segments for this case gives zero contribution which is unrealistic. The solution is to replace these two straight tip vortex-line segments with a circular arc segment passing through the three different end points of these segments. This circular arc segment has the same vortex core radius as the rest of the tip vortex. The geometry and the contribution to the induced velocity at point P_η of this circular arc segment are derived in Appendix C. The problem with the adjacent shed vortex sheet segments is the edge singularity associated with any vortex sheet. This is avoided by replacing these two adjacent vortex sheet segments with one big vortex sheet segment which has point P_η in the middle.

The information actually input to WG-71 from LDS-73 is the angle-of-attack distribution $\alpha(\eta,\psi)$ in degrees. The bound circulation distribution is obtained from:

$$\gamma(\eta,\psi) = \frac{a}{360}\left(\eta + \mu\sin\psi\right)\alpha(\eta,\psi) \tag{4.5}$$

which is analogous to Eq. 3.1. This bound circulation distribution is used directly for the n trailing vortex-line case, where n is the number of η stations plus one. Otherwise, the circulation models of Subsection 2.8 are used as necessary.

The vortex core radii used by WG-71 are normally the same as those used by LDS-73. A table of ages for tip-vortex core bursting is also obtained from LDS-73. There is no provision for vortex sheet vortex cores in WG-71, again because this is a recent addition to LDS-73 which did not make much difference in the final answers.

4.1.3 Wake-Geometry Extrapolation

The tip vortex distortion is only computed for the wake which is within $2\pi m$ of the rotor (i.e., for $0 \leq \delta \leq 2\pi m$). When computing the distortion $\vec{D}(\psi,\delta)$ at point P_η it is clearly necessary to consider the induced velocity contributions of all of the wake within $2\pi M$ of point P_η, where M is normally at least as large as m. Thus, δ_ℓ will range from 0 to $\delta + 2\pi M$ for any given δ and will have a maximum range of 0 to $2\pi(m + M)$. This implies a need to know the geometry of the tip vortex for anges δ_ℓ greater than $2\pi m$.

One solution to this problem is to simply neglect all wake for ages greater than $2\pi m$. This means that the $\vec{D}(\psi,\delta)$ calculated for δ near $2\pi m$ will be very inaccurate. This is due to the neglect of some nearby wake which can be expected to make a large contribution to the induced velocity at point P_η and hence to $\vec{D}(\psi,\delta)$. Thus, m must be approximately doubled so that the inaccurate $\vec{D}(\psi,\delta)$ computed for δ between $2\pi(m - M)$ and $2\pi m$ can be discarded and still leave an adequate amount of accurate $\vec{D}(\psi,\delta)$ near the rotor.

A much more economical and more accurate solution is to use extrapolation of the tip-vortex geometry for δ_ℓ greater than $2\pi m$. This allows all of the wake within $2\pi M$ in age of any point P_η to be included in a distortion computation, just as all of the wake within $2\pi m$ of any point P_η on the rotor is included in an airloads computation. This is the solution used by WG-71. The extrapolation scheme chosen assumes that the distortion parallel to the TPP remains constant for $\delta_\ell > 2\pi m$, while the distortion perpendicular to the TPP is computed, using a mean induced velocity for $\delta_\ell > 2\pi m$. This implies the following relations for $\delta_\ell > 2\pi m$:

$$D_1(\phi,\delta_\ell) = D_1(\phi, 2\pi m) \tag{4.6}$$

$$D_2(\phi,\delta_\ell) = D_2(\phi, 2\pi m) \tag{4.7}$$

$$D_3(\phi,\delta_\ell) = D_3(\phi, 2\pi m) + (\lambda - \mu \tan i)(\delta_\ell - 2\pi m) \tag{4.8}$$

where the components of $\vec{D}(\phi,\delta_\ell)$ are defined by Eq. 3.17 and λ is an input mean inflow, normally the same as for a rigid wake model.

4.1.4 Iteration Scheme

As discussed at the beginning of this section, it is necessary to know the distortion in order to be able to compute the distortion. This means that an iteration scheme is required. Any iteration scheme must also include provisions for starting the iteration. Various possible iteration and starting schemes are described in this subsection and the reasons for choosing the particular schemes used by WG-71 are indicated.

The basic requirement for starting the $\vec{D}$ iteration is an initial assumption for the wake geometry. One possibility is to assume that the rotor is instantaneously started from rest, which implies that there is initially no wake. As the rotor rotates new wake is generated and the tip-vortex distortion is computed for this new wake as it is generated. This scheme is analogous to the neglect of all wake for $\delta_\ell > 2\pi m$, as discussed in Subsection 4.1.3. The problem with this starting scheme is similar to the problem with the neglect of wake for $\delta_\ell > 2\pi m$. The $\vec{D}$ computed for the first $2\pi m$ or so in age is very inaccurate and extra computation is required to achieve a converged result.

The scheme used by WG-71 is to start with $2\pi m$ of assumed wake geometry. Normally, the rigid-wake assumption is used for this purpose. As the computation proceeds, the rotor rotates generating new wake. The tip-vortex distortion is computed for this new calculated wake, but not for the old assumed wake, which retains its original, assumed geometry. As new calculated wake is generated, the rotor moves away from the assumed wake. Eventually enough calculated wake ($2\pi m$ in age) is generated to provide either a calculated $\vec{D}$ or an extrapolation of the calculated $\vec{D}$ for all purposes and the assumed wake is no longer needed. This starting scheme gives rapid convergence and is recommended.

After the tip-vortex distortion computation is started by generating the first $2\pi m$ of new wake, a scheme for the rest of the computation is required. Two possibilities will be considered: continuous computation and iteration. For the continuous computation scheme, the rotor continues to advance generating new wake. However, only the distortion of the tip vortex within $2\pi m$ of the rotor is being computed. Therefore, every time the rotor generates $\Delta\psi$ of new calculated tip vortex ($\delta = 0$ to $\Delta\psi$), the oldest remaining

$\Delta\psi$ of calculated tip vortex is dropped ($\delta = 2\pi m$ to $2\pi m + \Delta\psi$) from the computation. Thus, the calculated table of $\vec{D}(\psi,\delta)$ is limited to $0 < \delta < 2\pi m$. As discussed in Subsection 4.1.3, an extrapolated $\vec{D}(\phi,\delta_\ell)$ is available for $\delta_\ell > 2\pi m$. This continuous computation continues until the calculated $\vec{D}(\psi,\delta)$ converges.

For the iterative scheme, no more new calculated wake is generated beyond the starting $2\pi m$. Instead a new iteration is begun. This new iteration starts with an assumed wake consisting of $2\pi m$ of distorted wake from the previous iteration plus $2\pi(M - m)$ of extrapolated distortion. The computation then proceeds as in the starting scheme, but with this revised initial wake-geometry assumption. The iteration is repeated until distortion convergence is achieved. Normally, two or three iterations are required. This iteration scheme is used by WG-71 because it is slightly more convenient for use with some of the refinements described in Subsection 4.2. Basically, either continuous computation or iteration can be used satisfactorily.

4.1.5 Basic Distortion Iteration

A simplified flow chart for the basic distortion iteration is given in Fig. 25. This flow chart represents one iteration of the fundamental distortion computation, the brute-force approach to tip vortex-geometry computation. The more complicated development of this technique used by WG-71 will be described in Subsection 4.2.

The first step is to initialize $\vec{D}_N(\psi)$ and δ_D which will be defined later. Now consider the outer loops on δ_M and δ. Advancing δ_M by $\Delta\psi$ causes the boundary between assumed and calculated wake to move forward by $\Delta\psi$ and it causes a segment $(\Delta\psi)$ of new calculated wake to be generated. Thus, δ_M starts at $\Delta\psi$, representing the first increment of new calculated wake. When $\delta_M = 2\pi m$, the $D(\psi,\delta)$ for $2\pi m$ of tip vortex has been calculated and the iteration is over. The δ loop is updating, except for the last step when $\delta = \delta_M$. For this brute force approach, complete updating is done at every step in δ_M. If no updating were done, the δ loop would be eliminated and δ would always be equal to δ_M.

The next loop is on ψ. Since both δ and ψ are now determined, it is possible to locate point P_η using the current best estimate of $\vec{D}(\psi,\delta)$ (see Subsection 4.1.6). The induced velocity totals $\Sigma\vec{q}$, $\vec{q}_{BV}$, and $\Delta\vec{q}$ are all set equal to zero here. These totals are accumulated for each point P_η and will be needed later. The total induced velocity at point P_η at age δ is $(\Sigma\vec{q} + \vec{q}_{BV})$, where $\vec{q}_{BV}$ is the contribution of the bound vorticity. The contribution of all the new wake generated by the last $\Delta\psi$ increment to δ_ℓ (i.e., the wake between $\delta_\ell = 0$ and $\delta_\ell = \Delta\psi$) to the induced velocity at point P_η is called $\Delta\vec{q}$.

The next loop is on ζ. This is done to include the contributions to the induced velocity at point P_η of the wake and the bound vorticity of each blade.

The next loop is on δ_ℓ. This represents integration up the wake from $\delta_\ell = \delta + 2\pi M$ to $\delta_\ell = 0$. Given ψ, ζ, δ, and δ_ℓ, the value of ϕ is determined by Eq. 4.3. Thus, the δ_ℓ loop can be thought of as a ϕ loop. When WG-71 was written this loop was considered a ϕ loop and the variable names in WG-71 reflect this. The loop will be explained here as a δ_ℓ loop, however, because this approach is much easier to understand.

Before the ℓ loop starts, another induced velocity total $\vec{q}_s$ is set equal to zero. This $\vec{q}_s$ is the contribution to the induced velocity at point P_η of all of the wake of one blade between the ages δ_ℓ and $\delta_\ell - \Delta\psi$. The last value of $\vec{q}_s$ (for $\delta_\ell = 0$) is the only one actually used by the basic distortion iteration. In fact, if the ζ loop were placed inside the δ_ℓ loop, $\Delta\vec{q}$ could be defined directly and $\vec{q}_s$ would not be needed here. However, all values of $\vec{q}_s$ will be needed by the near/far wake scheme (Subsection 4.2.1), so they are defined here to minimize the changes required.

The ℓ loop provides for the contributions to the induced velocity of the wake and bound vorticity associated with the various radial stations ℓ. Since ℓ, ϕ, and δ_ℓ are now all known, the location of point P_ℓ can be determined using the current best estimate of $\vec{D}(\phi,\delta_\ell)$ (see Subsection 4.1.6). The vectors $\vec{a}$ and $\vec{b}$ connecting point P_η with the various points P_ℓ can now be determined. This data can then be used to compute the actual contributions to the induced velocity at point P_η from the various wake elements ($\vec{q}$) and, when $\delta_\ell = 0$, from the bound vorticity ($\Delta\vec{q}_{BV}$). The details of computing $\vec{a}$, $\vec{b}$, $\vec{q}$, and $\vec{q}_{BV}$ will be covered in Subsections 4.1.7 and 4.1.8. The computed $\vec{q}$ is added to the totals $\Sigma\vec{q}$ and $\vec{q}_s$. If $\delta_\ell = 0$, then $\Delta\vec{q}_{BV}$ is added to $\vec{q}_{BV}$. This ends the ℓ loop.

Working outward through the nest of loops, δ_ℓ and ϕ are changed by $\Delta\psi$ and a test is made for the end of the δ_ℓ loop. At the end of the δ_ℓ loop, $\delta_\ell = 0$, which implies that the current value of $\vec{q}_s$ can be added to $\Delta\vec{q}$. Because this is inside the ζ loop, there will be a contribution to $\Delta\vec{q}$ from each blade as ζ changes.

When the ζ loop is finished, the induced velocity at point P_η has been computed. This has involved summing contributions from various radial stations (ℓ) and various blades (ζ) as well as an integration over the azimuthal extent of the wake (δ_ℓ). Now the average induced velocity at point P_η between ages $\delta - \Delta\psi$ and δ is:

$$\vec{q}_0(\psi) = \Sigma \vec{q} + \vec{q}_{BV} - 0.5 \Delta \vec{q} \tag{4.9}$$

where $\vec{q}_{BV}$ has already been averaged (see Subsection 4.1.7).

At this point the main ψ loop ends and the computation of $\vec{D}(\psi,\delta)$ occurs inside a separate ψ loop. This is done to avoid the problem of trying to use a table of distortion where the boundary between computed $\vec{D}$ and assumed $\vec{D}$ is δ for azimuth angles less than ψ, and $\delta - \Delta\psi$ for azimuth angles greater than or equal to ψ. Before a new $\vec{D}(\psi,\delta)$ is computed, the old value is stored as $\vec{D}_O(\psi)$ and the current value of δ is stored as δ_D. Now, the incremental distortion is computed as the averaged induced velocity times the time increment ($\Delta\psi$):

$$\vec{D}(\psi,\delta) = \vec{D}(\psi,\delta - \Delta\psi) + \Delta\psi\, \vec{q}_0(\psi) \tag{4.10}$$

To start this incremental computation of $\vec{D}(\psi,\delta)$, a value for $\vec{D}(\psi,0)$ is required. At age $\delta = 0$, there has been zero time for the wake to accumulate distortion since leaving the rotor blade; therefore:

$$\vec{D}(\psi,0) = 0.0 \tag{4.11}$$

The δ loop ends when $\delta = \delta_M$.

After the boundary age, δ_M, is incremented, the assumed value of $\vec{D}(\psi, \delta_M - \Delta\psi)$ is moved from $\vec{D}_N(\psi)$ to $\vec{D}_S(\psi)$ and the assumed value of $\vec{D}(\psi,\delta_M)$ is stored in $\vec{D}_N(\psi)$. The only function of $\vec{D}_N(\psi)$ is as an intermediary between $\vec{D}$ and $\vec{D}_S$. For the basic distortion iteration $\vec{D}_O$ could be used instead of $\vec{D}_N$, but this is not true for WG-71. The function of $\vec{D}_O(\psi)$ and $\vec{D}_S(\psi)$ is to aid in the location of point P_η (see Subsection 4.1.6).

The δ_M loop ends when $\delta_M = 2\pi m$. At this point the distortion of the $2\pi m$ of wake nearest the rotor has been computed and an iteration has been completed. There is no test for convergence of the distortion computation in WG-71. Instead, the number of iterations is an input (normally two is sufficient). Distortion convergence is determined by examination of tip vortex geometry plots for successive iterations.

The computation is arranged such that only one complete $\vec{D}$ table is required. At the start of an iteration, this table contains the assumed $\vec{D}$. As the computation proceeds, the computed $\vec{D}$ steadily replaces the assumed $\vec{D}$ in the table. The boundary between the computed $\vec{D}$ and the assumed $\vec{D}$ is age δ_M. At the end of an iteration, the table contains only computed $\vec{D}$. Recall that at the start of a new iteration, the computed $\vec{D}$ from the previous iteration becomes the assumed $\vec{D}$ for the new iteration. Thus the $\vec{D}$ table does not need to be changed to start a new iteration.

4.1.6 Location of Points P_ℓ and P_η

The location of points P_ℓ and P_η requires careful consideration because it is important to use all of the available information to achieve the best possible estimate. The basic problem is that the best estimate of the distortion $\vec{D}$ changes as a distortion iteration proceeds. Furthermore, in the middle of a distortion iteration, the $\vec{D}$ for the computed wake is more accurate than the $\vec{D}$ for the assumed wake, since it has been computed more recently. The estimation of the distortion at points P_ℓ and P_η is therefore more complicated than simply using $\vec{D}(\phi,\delta_\ell)$ and $\vec{D}(\psi,\delta)$.

Consider the state of the $\vec{D}$ table at a point in the middle of an iteration. The table will have three distinct areas containing different types of $\vec{D}$. For ages greater than or equal to δ_M, the table will contain assumed $\vec{D}$. This is the least accurate type of $\vec{D}$ since it is the oldest. For ages less than or equal to δ_D, the table will contain computed values of $\vec{D}$ which have just been updated. This is the most accurate type of $\vec{D}$ since it is the most recently computed. For ages greater than δ_D, but less than δ_M, the table will contain computed $\vec{D}$ values which have not been updated as recently as the $\vec{D}$ of the preceding section, if it has been updated at all. This intermediate section of the $\vec{D}$ table will not exist if δ_D happens to be equal to δ_M.

While $\vec{D}(\psi,\delta)$ describes the distortion at point P_η, it is not always the most accurate available description. Define $\vec{D}_\eta(\psi,\delta)$ as the best possible estimate for the distortion at point P_η. Then for $\delta \leq \delta_D$:

$$\vec{D}_\eta(\psi,\delta) = \vec{D}(\psi,\delta) \tag{4.12}$$

For $\delta_D < \delta < \delta_M$:

$$\vec{D}_\eta(\psi,\delta) = \vec{D}(\psi,\delta_D) + \vec{D}(\psi,\delta) - \vec{D}_o(\psi) \tag{4.13}$$

and for $\delta \geq \delta_M$:

$$\vec{D}_\eta(\psi,\delta) = \vec{D}(\psi,\delta_D) + \vec{D}(\psi,\delta_M) - \vec{D}_o(\psi) + \vec{D}(\psi,\delta) - \vec{D}_s(\psi) \tag{4.14}$$

Consider the case $\delta \geq \delta_M$ (Eq. 4.14). For this case, the distortion accumulated up to age δ_D is estimated by $\vec{D}(\psi, \delta_D)$, which has just been updated. The distortion accumulated between age $(\delta_D + \Delta\psi)$ and age $(\delta_M - \Delta\psi)$ is estimated by $\vec{D}(\psi, \delta_M) - \vec{D}_O(\psi)$, which is calculated distortion not updated as recently as $\vec{D}(\psi,\delta_D)$. Finally, the distortion accumulated between age δ_M and δ is estimated by $\vec{D}(\psi,\delta) - \vec{D}_s(\psi)$, which is assumed distortion.

For point P_ℓ the same technique is used. Define $\vec{D}_\ell(\phi,\delta_\ell)$ as the best available estimate of the distortion at point P_ℓ. Then for $\delta_\ell \leq \delta_D$:

$$\vec{D}_{\ell}(\phi, \delta_{\ell}) = \vec{D}(\phi, \delta_{\ell}) \tag{4.15}$$

for $\delta_D < \delta_\ell < \delta_M$:

$$\vec{D}_{\ell}(\phi, \delta_{\ell}) = \vec{D}(\phi, \delta_{D}) + \vec{D}(\phi, \delta_{\ell}) - \vec{D}_{0}(\phi) \tag{4.16}$$

and for $\delta_\ell \geq \delta_M$:

$$\vec{D}_{\ell}(\phi, \delta_{\ell}) = \vec{D}(\phi, \delta_{D}) + \vec{D}(\phi, \delta_{M}) - \vec{D}_{0}(\phi) + \vec{D}(\phi, \delta_{\ell}) - \vec{D}_{s}(\phi) \tag{4.17}$$

Extrapolation is never necessary to find $\vec{D}_\eta(\psi,\delta)$ since $\delta \leq 2\pi m$. However, δ_ℓ can exceed $2\pi m$ and in this case, extrapolation becomes necessary for $\vec{D}_\ell(\phi,\delta_\ell)$, (see Subsection 4.1.3). The extrapolation technique includes computing $\vec{D}_\ell(\phi,\delta_\ell)$ for $\delta = 2\pi m$ and then adding $(\lambda - \mu \tan i)(\delta_\ell - 2\pi m)$ to the $\vec{k}$ component of $\vec{D}_\ell(\phi, 2\pi m)$, (see Eqs. 4.6, 4.7, and 4.8).

Once the distortions $\vec{D}_\eta(\psi,\delta)$ and $\vec{D}_\ell(\phi,\delta_\ell)$ have been evaluated, the locations of points P_η and P_ℓ are easily found. For distortion computations, the absolute locations of points P_ℓ and P_η are not needed. What is needed is the vector ($\vec{a}$ or $\vec{b}$) connecting point P_η with point P_ℓ. The formulas for the vector components of $\vec{a}$ and $\vec{b}$ will be developed when the computation of the various contributions to the induced velocity is described (Subsections 4.1.7 and 4.1.8).

4.1.7 Wake Contribution to Induced Velocity

The computation of the induced velocity contribution of the wake in WG-71 is very similar to the induced velocity computation in LDS-73 (Subsections 3.2.3, 3.2.5, and 3.2.6). The differences between these computations are due to slightly different wake models (Subsection 4.1.2) and due to differences in the locations of points P_ℓ and P_η. Point P_η is located on a rotor blade for airloads computations and has no distortion. For distortion computations, point P_η is located on the tip vortex. Thus, $\vec{D}_\eta(\psi,\delta)$

must be included in distortion computations. At point P_ℓ the distortion $\vec{D}(\phi,\delta)$ is simply looked up in the table of $\vec{D}$ for airloads computations. For distortion computations $\vec{D}_\ell(\phi,\delta_\ell)$ must be used (Subsection 4.1.6). It is also important to recall that the notation δ has different meanings in WG-71 and LDS-73 (Subsection 4.1.1).

The vectors $\vec{a}$ and $\vec{b}$ are vectors beginning at point P_η and ending at various points P_ℓ (which are labeled points P_a and P_b, as necessary).

Recall the components of $\vec{a}$ and $\vec{b}$:

$$\vec{a} = x_a \vec{i} + y_a \vec{j} + z_a \vec{k} \tag{3.9}$$

$$\vec{b} = x_b \vec{i} + y_b \vec{j} + z_b \vec{k} \tag{3.10}$$

where $\vec{i}$, $\vec{j}$, and $\vec{k}$ are defined in Subsection 3.2.3. Define the components of $\vec{D}_\eta(\psi,\delta)$ and $\vec{D}_\ell(\phi,\delta_\ell)$ as:

$$\vec{D}_\eta(\psi,\delta) = D_{\eta x}(\psi,\delta)\,\vec{i} + D_{\eta y}(\psi,\delta)\,\vec{j} + D_{\eta z}(\psi,\delta)\,\vec{k} \tag{4.18}$$

$$\vec{D}_\ell(\phi,\delta_\ell) = D_{\ell x}(\phi,\delta_\ell)\,\vec{i} + D_{\ell y}(\phi,\delta_\ell)\,\vec{j} + D_{\ell z}(\phi,\delta_\ell)\,\vec{k} \tag{4.19}$$

Now consider the trailing wake. Point P_a is at the end of the vortex line (or vortex sheet) segment having the smallest age. Thus, point P_a is identified by (ℓ,ϕ,δ_ℓ) while point P_b is identified by $(\ell,\ \phi - \Delta\psi,\ \delta_\ell + \Delta\psi)$. The equations for the components of $\vec{a}$ and $\vec{b}$ are analogous to Eqs. 3.18 through 3.23.and are derived the same way:

$$x_a = \ell \sin\phi - \eta \sin\psi + D_{\ell x}(\phi,\delta_\ell) - D_{\eta x}(\psi,\delta) \tag{4.20}$$

$$x_b = \ell\sin(\phi-\Delta\psi) - \eta\sin\psi + D_{\ell x}(\phi-\Delta\psi,\delta_\ell+\Delta\psi) - D_{\eta x}(\psi,\delta) \tag{4.21}$$

$$y_a = \ell\cos\phi - \eta\cos\psi + \mu(\delta_\ell-\delta) + D_{\ell y}(\phi,\delta_\ell) - D_{\eta y}(\psi,\delta) \tag{4.22}$$

$$\begin{aligned} y_b = {} & \ell\cos(\phi-\Delta\psi) - \eta\cos\psi + \mu(\delta_\ell+\Delta\psi-\delta) \\ & + D_{\ell y}(\phi-\Delta\psi,\delta_\ell+\Delta\psi) - D_{\eta y}(\psi,\delta) \end{aligned} \tag{4.23}$$

$$z_a = (\delta_\ell-\delta)\mu\tan i - a_0(\ell-\eta) + D_{\ell z}(\phi,\delta_\ell) - D_{\eta z}(\psi,\delta) \tag{4.24}$$

$$\begin{aligned} z_b = {} & (\delta_\ell+\Delta\psi-\delta)\mu\tan i - a_0(\ell-\eta) \\ & + D_{\ell z}(\phi-\Delta\psi,\delta_\ell+\Delta\psi) - D_{\eta z}(\psi,\delta) \end{aligned} \tag{4.25}$$

For the tip vortex, η is unity. The factor $(\delta_\ell - \delta)$ is written as $(\psi + \zeta - \phi)$ in WG-71, which is equivalent (see Eq. 4.3).

For the tip vortex line, case γ_a is computed using Subsection 2.8.1. The induced velocity contribution is:

$$\vec{q} = \gamma_c(\rho)\ \vec{q}_{VL} \tag{4.26}$$

for all δ_ℓ, except (when $\zeta = 0$), $\delta_\ell = \delta$, and $\delta_\ell = \delta - \Delta\psi$. The vortex core factor $\gamma_c(\rho)$ is given by Eq. 3.4.1 and $\vec{q}_{VL}$ is given by Eq. A.24. The vortex core radius ρ_c is the same as that for airloads computations, including vortex core bursting. Since point P_η is in the wake, not on a rotor blade, the other close blade-vortex interaction effects do not apply.

The two tip vortex-line-segments adjacent to point P_η are represented by a single curved vortex-line-segment instead of two straight-line segments (Subsection 4.1.2). These segments are both shed from the same blade as point P_η; i.e., $\zeta = 0$, and have ages $\delta_\ell = \delta$ and $\delta_\ell = \delta - \Delta\psi$. Recall from Subsection 1.5.2, that the age of a trailing vortex-line segment is defined as the age of its newer (smaller age, larger azimuth angle) end. Thus, points P_a and P_b for the single curved tip vortex-line segment are identified by (ϕ, δ_ℓ) and $(\phi - 2\Delta\psi, \delta_\ell + 2\Delta\psi)$, respectively, where $\delta_\ell = \delta - \Delta\psi$ and $\phi = \psi + \zeta + \Delta\psi$. The components of $\vec{a}$ are computed using Eqs. 4.20, 4.22, and 4.24 directly. The components of $\vec{b}$ are computed using Eqs. 4.21, 4.23, and 4.25, but with $\Delta\psi$ replaced by $2\Delta\psi$. The vortex core radius ρ_c and circulation γ_a remain the same as for the straight-line-segment case. The $\vec{q}$ contribution is equal to $\vec{q}_{CVL}(\Delta\theta_a + \Delta\theta_b)$, which is computed using Eq. C.37. The $\vec{a}_c$ and $\vec{b}_c$ of Appendix C are equal to $\vec{a}$ and $\vec{b}$, respectively. Vortex-core effects are included in Eq. C.37.

For the single inboard trailing vortex line model, γ_a is computed using Subsection 2.8.1, and the vortex core radius is an input ρ_I. The components of $\vec{a}$ and $\vec{b}$ are computed using Eqs. 4.20 through 4.25 with $\ell = \ell_2$ (Subsection 4.1.2). The contribution to $\vec{q}$ is obtained from Eq. 4.26.

For the inboard trailing vortex sheet case, $\vec{a}$ and $\vec{b}$ are also computed using Eqs. 4.20 through 4.25 with $\ell = \ell_2$, and γ_a is defined in Subsection 2.8.1. No vortex core radius is used, but z_m and $\Delta\tilde{x}$ are required (see Appendix B):

$$\Delta\tilde{x} = \ell_2 - \ell_1 \tag{4.27}$$

where ℓ_1 is the input root cutout, and

$$z_m = 0.5\,(z_a + z_b) \tag{4.28}$$

where z_a and z_b are evaluated for $\ell = \ell_1$, using Eqs. 4.24 and 4.25.

The contribution to $\vec{q}$ is equal to $\vec{q}_{VS}$, which is computed using Eqs. B.22 or B.23, as necessary.

The n trailing-vortex-line case is not included in Subsection 2.8.1 so the computation of γ_a will be described here. Label the $n_\ell = n$ trailing vortices ℓ_ℓ, where ℓ_ℓ runs from ℓ_1 at the root cutout to $\ell_n = 1.0$ at the blade tip. Label the $n_\eta = n - 1$ radial stations at which $\gamma_{bv}(\eta_k, \phi)$ is known as η_k, where η_k runs from η_1 between ℓ_1 and ℓ_2 to η_{n-1} between ℓ_{n-1} and ℓ_n. Let the circulation of the trailing vortex line ℓ_ℓ be $\gamma_t(\ell_\ell, \phi)$. Now at the root ℓ_1 and at the blade tip $\ell_n = 1.0$,

$$\gamma_t(\ell_1, \phi) = -\gamma_{bv}(\eta_1, \phi) \tag{4.28}$$

$$\gamma_t(1.0, \phi) = \gamma_{bv}(\eta_{n-1}, \phi) \tag{4.29}$$

For all intermediate ℓ_ℓ,

$$\gamma_t(\ell_\ell, \phi) = \gamma_{bv}(\eta_{\ell-1}, \phi) - \gamma_{bv}(\eta_\ell, \phi) \tag{4.30}$$

For all ℓ_ℓ, including ℓ_n, the tip vortex,

$$\gamma_a = \gamma_t(\ell_\ell, \phi) \tag{4.31}$$

Finally, the contribution to $\vec{q}$ is computed using Eq. 4.26.

Now, consider the shed-wake case. For this case, $\vec{a}$ is the same as for the tip-vortex case and it is computed using Eqs. 4.20, 4.22, and 4.24. Point P_b is at the same azimuth ϕ and age δ_ℓ as point P_a, but at a different ℓ. Thus, the components of $\vec{b}$ are also computed using Eqs. 4.20, 4.22, and 4.24, but with a different ℓ. For the basic wake model with a single inboard trailing-vortex line, point P_a is at ℓ_3, and point P_b is at ℓ_2. If an inboard trailing-vortex sheet model is used together with a vortex-line-shed-wake model, point P_b moves to ℓ_1. For the n trailing-vortex-line case, the shed-wake segment associated with ℓ_ℓ has point P_a at ℓ_ℓ and point P_b at $\ell_{\ell-1}$. For all shed-vortex-line cases, the vortex-core radius is an input ρ_L. Normally γ_a and γ_b are equal to $\gamma_s(\phi)$ from Eq. 2.9; however, when the inboard trailing vortex-sheet model is used, the shed-vortex-line circulation varies from $\gamma_a = \gamma_s(\phi)$ to $\gamma_b = 0.0$. Finally, the contribution to $\vec{q}$ is computed using Eq. 4.26 where $\vec{q}_{VL}$ is obtained from Eq. A.23 or A.24, as necessary.

For the shed-vortex-sheet case there is always a segment having point P_a at ℓ_3 and point P_b at ℓ_2. For this segment both γ_a and γ_b are equal to $\gamma_s(\phi)$ from Eq. 2.9. There is no vortex-core radius but $\Delta\tilde{x}$ and z_m are needed. To compute these parameters define $\vec{a}_s$ and $\vec{b}_s$ as $\vec{a}$ and $\vec{b}$, now evaluated at $\phi + \Delta\psi$ instead of ϕ and $\delta_\ell - \Delta\psi$ instead of δ_ℓ. Now,

$$\Delta\tilde{x} = 0.5\left(|\vec{a}_s - \vec{a}| + |\vec{b}_s - \vec{b}|\right) \tag{4.31}$$

$$z_m = 0.5\left(z_{as} + z_{bs}\right) \tag{4.32}$$

where z_{as} and z_{bs} are the $\vec{k}(z)$ components of $\vec{a}_s$ and $\vec{b}_s$, respectively. The contribution to $\vec{q}$ is equal to $\vec{q}_{VS}$ which is computed using Eq. B.23.

When an inboard trailing vortex-sheet model is used, there is a second shed-vortex-sheet segment. This segment has point P_a at ℓ_2 and P_b at ℓ_1. The circulation varies from $\gamma_a = \gamma_s(\phi)$ to $\gamma_b = 0.0$. The parameters $\Delta\tilde{x}$ and z_m are obtained using Eq. 4.31 and 4.32 with the proper values of ℓ. The contribution to $\vec{q}$ is $\vec{q}_{VS}$ computed using Eq. B.22.

The two shed vortex-sheet segments adjacent to point P_η are represented by one large segment to avoid the edge singularity (Subsection 4.2.1). These segments are both $\zeta = 0$ and have ages $\delta_\ell = \ell$ and $\delta_\ell = \delta + \Delta\psi$. Thus the single large segment has $\vec{a}$ and $\vec{b}$ defined at age $\delta_\ell = \delta + \Delta\psi$ and azimuth angle $\phi = \psi + \zeta - \Delta\psi$, while a_s and b_s are defined at age $\delta_\ell = \delta - \Delta\psi$ and azimuth angle

$\phi = \psi + \zeta + \Delta\psi$. The rest of the computation proceeds like that for a normal shed-vortex-sheet segment.

4.1.8 Bound Vorticity Contribution to Induced Velocity

Assuming lifting-line theory and steady-state blade motion, the net contribution of the bound vorticity to the induced velocity at points on the rotor blades is zero. Thus, the contribution of the bound vorticity to the induced velocity is neglected in airloads computations, except for the use of lifting-surface theory during a close blade-vortex interaction (Subsection 3.2.4). The bound vorticity cannot be neglected in distortion computations, however, because the induced velocity is computed at points in the wake.

Normally, the bound vorticity is represented by one or more vortex-line segments. These bound vortex-line segments can be thought of as the last of the set of shed-vortex-line segments. While the shed-vortex-line segments have ages δ_ℓ running from $\delta + 2\pi M$ to $\Delta\psi$ the age of the bound vorticity is always zero ($\delta_\ell = 0$). The sign convention for $\gamma_{bv}(\phi)$ is the opposite of the sign convention for $\gamma_s(\phi)$. Therefore points P_a and P_b are interchanged for the bound vorticity compared to the shed wake. Thus, point P_a is always inboard of point P_b in bound vorticity computations. Naturally, γ_a is equal to $\gamma_{bv}(\phi)$ or $\gamma_{bv}(\ell_\ell,\phi)$ as necessary. Keeping the differences listed here in mind, the contribution of the bound vorticity to the induced velocity ($\Delta\vec{q}_{BV}$) is computed using the same techniques as for the corresponding shed-wake-vortex-line cases (Subsection 4.1.7).

While the technique outlined above is normally satisfactory, there is a problem when the bound vortex line passes near point P_η. The rotor blade is indexed in age in increments of $\Delta\psi$ (typically $\Delta\psi = 15°$). When the rotor blade is close to point P_η a $\Delta\psi$ change in the location of the blade can make a very large change in $\Delta\vec{q}_{BV}$.

Thus, sampling $\Delta\vec{q}_{BV}$ at $\Delta\psi$ intervals is not very accurate when the blade is close to point P_η.

The object of computing $\Delta\vec{q}_{BV}$ is to ultimately find its contribution to the distortion $\vec{D}(\psi,\delta)$. Ideally, this should be done by integrating $\Delta\vec{q}_{BV}$ over age δ, as age goes from an initial value $\delta - \Delta\psi$ to δ; thus obtaining the contribution to $\vec{D}(\psi,\delta)$ directly. The program WG-71 is not able to compute $\vec{D}(\psi,\delta)$ at this point, so the integrated $\Delta\vec{q}_{BV}$ is divided by $\Delta\psi$ to get an average value for $\Delta\vec{q}_{BV}$ between ages $\delta - \Delta\psi$ and δ. Later this will be multiplied by $\Delta\psi$ to get the increment to the distortion back again (Eq. 4.10).

Now recall that $\delta_\ell = 0$ for the bound vorticity. Thus, as age goes from $\delta - \Delta\psi$ to δ the azimuth angle ϕ of the bound vortex line goes from $\psi + \zeta + \delta - \Delta\psi$ to $\psi + \zeta + \delta$. Therefore, integrating $\Delta\vec{q}_{BV}$ over age from $\delta - \Delta\psi$ to δ and dividing the result by $\Delta\psi$ is the same mathematically (but not physically) as spreading the bound vortex line out into a vortex sheet running from $\phi = \psi + \zeta + \delta - \Delta\psi$ to $\phi = \psi + \zeta + \delta$. This is convenient since it allows use of the vortex sheet formulas already built into WG-71. However, it is important to understand that the bound vortex line remains a line not a sheet. All that has been done is to integrate $\Delta\vec{q}_{BV}$ over age δ (which is equivalent to integration over azimuth angle ϕ for this special case), and divide the result by $\Delta\psi$ to obtain an average $\Delta\vec{q}_{BV}$.

The vectors $\vec{a}$ and $\vec{b}$ are the same as for a normal bound-vortex-line computation. To find $\Delta\tilde{x}$ and z_m define $\vec{a}_{BV}$ and $\vec{b}_{BV}$ as $\vec{a}$ and $\vec{b}$ evaluated for age $\Delta\psi$ instead of zero and azimuth angle $\psi + \zeta + \delta - \Delta\psi$ instead of $\psi + \zeta + \delta$. Now:

$$\Delta\tilde{x} = 0.5\left(|\vec{a}_{BV} - \vec{a}| + |\vec{b}_{BV} - \vec{b}|\right) \tag{4.33}$$

$$z_m = 0.5\left(z_{aBV} + z_{bBV}\right) \tag{4.34}$$

where z_{aBV} and z_{bBV} are the $\vec{k}(z)$ components of $\vec{a}_{BV}$ and $\vec{b}_{BV}$, respectively. The induced velocity contribution $\Delta\vec{q}_{BV}$ is equal to $\vec{q}_{VS}$ computed using Eq. B.23.

The computation of this integrated $\Delta\vec{q}_{BV}$ is much more expensive than the computation of the normal $\Delta\vec{q}_{BV}$. Fortunately, use of the integrated $\Delta\vec{q}_{BV}$ is only important when the rotor blade passes near point P_η making $\Delta\vec{q}_{BV}$ large. Therefore, $\Delta\vec{q}_{BV}$ is always computed in the normal way. Then, if the resulting $\Delta\vec{q}_{BV}$ is greater than an input QBM, the $\Delta\vec{q}_{BV}$ is recomputed using the integration technique.

4.2 Distortion Computation Refinements

4.2.1 Near/Far Wake

Consider computing the contribution of a wake element to the induced velocity at point P_η. The farther away this wake element is from point P_η the less accuracy is required in specifying the geometry of the wake element relative to point P_η. This is because far wake elements make a smaller contribution to the induced velocity than near ones (assuming equal circulation), and because a given magnitude of error in the geometry will cause a percentage error in the geometry (and hence the induced velocity) which is smaller for far wake elements than for near ones. This suggests that the contribution of the far wake to the induced velocity at point P_η can be updated less often than the contribution of the near wake, without a significant loss in accuracy.

The basic near/far wake concept is that the contribution of each wake element to the induced velocity at each point P_η is computed at least once. After this initial computation,each wake element is assigned to either the near or the far wake, relative to each point P_η, based upon the magnitude of its contribution to the induced velocity. Any wake element whose contribution is greater than DM (an input to WG-71) becomes part of the near wake, while the rest of the wake becomes the far wake. Clearly, each different point P_η will have a different near wake and far wake. A particular wake element can be part of the near wake for some points P_η, and part of the far wake for other points P_η.

The problem with this basic near/far wake scheme is the amount of core storage (fast access memory) required to store the induced velocity contribution of each wake element at each point P_η. First consider the number of different points P_η. This is the number of azimuth angle stations ψ(25 for $\Delta\psi = 15°$) times the number of age stations δ(maximum of 97 in WG-71). This leads to 2425 points P_η which clearly leads to difficulties.

Consider the difference between ψ and δ. At different azimuth angles ψ the whole pattern of the wake is different. At different ages δ, with ψ held constant, the fundamental wake geometry remains fixed and only the distortion changes. Thus, those particular wake elements which compose the near wake and those which compose the far wake will change greatly as ψ changes, but will change very little as δ changes. Therefore, a different near wake and far wake is provided for each point P_η having a different azimuth angle ψ, but the division of the wake into near and far wakes remains the same for all ages δ. This provides nearly two orders-of-magnitude reduction in core storage (25 azimuth angles ψ instead of 2425 points P_η).

This assumption that the near/far wake division is independent of age δ is equivalent to saying that the near/far wake division is a function of age δ_R measured from point P_η, not age δ_ℓ measured from the rotor. Consider a wake element associated with point P_ℓ. Normally, point P_ℓ is identified by (ℓ,ϕ,δ_ℓ) where δ_ℓ is age measured from the rotor. However, consider the problem of assigning point P_ℓ to the near wake or to the far wake. It has been assumed that this assignment is independent of the age δ of point P_η. This means that it is also independent of the position of the rotor, since this position will change as δ changes. Since δ_ℓ is also measured from the position of the rotor, it is not suitable for identifying point P_ℓ in this case. Instead, the age of point P_ℓ measured relative to point P_η is appropriate.

Define this relative age as

$$\delta_R = \delta_\ell - \delta \tag{4.35}$$

where δ_R runs from $2\pi M$ down to $-\delta$ in steps of $\Delta\psi$. Notice that while δ and δ_ℓ are both functions of the position of the rotor, δ_R is not. To find the azimuth angle ϕ of point P_ℓ, substitute δ_R into Eq. 4.3:

$$\phi = \psi + \delta - \delta_R \tag{4.36}$$

Now the location of point P_η can be identified by (ℓ,ϕ,δ_R) instead of (ℓ,ϕ,δ_ℓ).

Although the near/far wake assignment of a wake element associated with a particular point P_η is independent of age δ, the existence of the wake element is not. As an example of this, observe that the lower limit of δ_R is $-\delta$. Thus, the contribution of either the near wake or the far wake to the induced velocity at point P_η will depend upon age δ because the number of wake elements which actually exist depends upon δ. This implies that two separate types of information must be stored:

1) the near/far wake assignment of each wake element relative to each azimuth angle ψ.
2) the total contribution of all far wake elements to the induced velocity at each point P_η.

The induced velocity contribution of the near wake does not require special storage provisions since it is recomputed every time it is needed.

The storage of the near/far wake assignments will be considered below, starting with the total number of wake elements. To store the far wake induced velocity contributions, define them as $\vec{q}_F(\psi,\delta)$. This has the same dimensions as $\vec{D}(\psi,\delta)$ and requires the same amount of core storage (6.8k bytes in WG-71, where one k is $2^{10} = 1024$). As will be seen below, this 6.8k bytes storage penalty is a minor penalty for the two order of magnitude reduction in near/far wake assignment storage.

Now consider the number of different wake elements. At each relative age δ_R there are n_ℓ trailing wake elements (maximum of 6 in WG-71) plus n_ℓ -1 shed wake elements. Now multiply this number by n_b blades (maximum of 6 in WG-71) and by the number of relative age δ_R stations (average value 1.5 times the number of δ stations). This product comes to 9603, and when further multiplied by 25 points P_η and 4 bytes per word requires about 940k bytes of core storage. For comparison, the entire WG-71 program requires 200 bytes, and when this program was originally developed, only about 400k bytes were available at MIT. Since then the amount of core storage available has increased by more than an order of magnitude; however, it is still costly.

Observe that when counting the number of wake elements, each point P_ℓ was identified by (ℓ,ζ,δ_R) instead of (ℓ,ϕ,δ_R). This is equivalent because ϕ can be obtained from Eq. 4.36 given ψ, ζ, and δ_R. The use of ζ instead of ϕ is desirable because there are normally fewer rotor blades (maximum of 6 in WG-71) than azimuth stations (25 for $\Delta\psi = 15°$). This suggests that attempts to reduce the number of ℓ stations and the number of δ_R stations may be profitable. First ℓ will be considered.

For a given age relative δ_R and rotor blade ζ all of the wake elements at the different radial stations ℓ are in the same general area. This suggests that all of these wake elements could be considered together as a unit. To do this, the contributions to the induced velocity at point P_η of all of these wake elements are added together. Then, if the total ($\vec{q}_s$ in Fig. 25) is greater than DM, all of these wake elements are assigned to the near wake. Otherwise all of these elements become part of the far wake. This eliminates ℓ and reduces the core storage required by a factor of 11 in WG-71. However, this saving in core storage is purchased at the cost of some extra computational expense. This is because some wake elements included in the near wake under this scheme would not have been included if they had been considered separately.

Due to the reduction in the cost of core storage relative to the cost of computation since WG-71 was originally developed, the developer of a next generation distortion computation program should investigate considering each wake element separately. The core storage penalty for this could be reduced from the factor of 11 in WG-71 to a factor of 3 by eliminating the n trailing vortex-line-wake model and using only the basic wake model as in LDS-73.

Now consider the reduction of the number of δ_R stations. It can be anticipated that a far wake element will tend to be adjacent to other far-wake elements more often than it is adjacent to near wake elements. Thus, as δ_R changes with ζ held constant there will be fewer transitions from far wake to near wake or from near wake to far wake than there are δ_R stations. This suggests that, instead of storing the near/far wake assignments for each group of wake elements having the same ζ and δ_R, only the δ_R at which these transitions occur needs to be stored.

Define the transition age $\delta_T(\zeta,\psi,k)$ as the relative age δ_R of the k'th transition in the wake of the rotor blade identified by ζ, relative to all points P_η having azimuth angle ψ. Large $\delta_T(\zeta,\psi, k)$ are associated with small k. Assuming that the wake at $\delta_R = 2\pi M$ will always be part of the far wake (if it is not, M is too small), then the first transition (k = 1) will be from far wake to near wake. Therefore, odd values of k will always be associated with transitions from far wake to near wake, and even values of k will be associated with transitions from near wake to far wake.

In WG-71 there are provisions for a maximum of 16 different k. Therefore, the storage required for $\delta_T(\zeta, \psi, k)$ is 6ζ times 25ψ times 16 k times 4 bytes per word which equals 9.4k bytes. The storage of δ_T instead of δ_R saves a factor of 9 in core storage and when combined with the factor of 11 saving due to summing over ℓ, this is another two order of magnitude saving, from 940k bytes to 9.4k bytes.

Now consider the actual use of this near/far wake scheme in WG-71. For this program, the induced velocity contribution of the far wake is only computed once per iteration. However, both the near/far wake assignments and the induced velocity contribution of the far wake are computed separately for each iteration. A program which used continuous computation instead of iteration (see Subsection 4.1.4) would presumably also update the far wake at about $2\pi m$ intervals in age.

To illustrate the details of using the near/far wake scheme, its application to the basic distortion iteration (Fig. 25) will be considered. The near/far wake scheme primarily affects the δ_ℓ loop, both at the start and at the finish. There are two basic cases to be considered:

1) The first time that the induced velocity contribution of a particular part of the wake is computed during the current iteration (for the basic distortion iteration this is whenever $\delta = \delta_M$).
2) During an update.

First, consider case (1). The age δ_ℓ will be computed as in Fig. 25b, since the contribution of every wake element must be computed the first time through the wake. Now the usefulness of $\vec{q}_s$ becomes clear. It is the sum over ℓ of the induced velocity contributions of the wake of one blade having age δ_ℓ. Thus, if $|\vec{q}_s| < DM$, this wake is assigned to the far wake. This means that $\vec{q}_s$ must be added to the far wake induced velocity contribution

$\vec{q}_F(\psi,\delta)$, which will be considered in more detail below. It is also necessary to test for a transition.

To aid in finding the next transition, define $k_m(\zeta,\psi)$ as the maximum value of the counter k in $\delta_T(\zeta,\psi,k)$ which has been achieved at this point in the distortion iteration. At the start of the distortion iteration $k_m(\zeta,\psi)$ is set equal to zero for all ζ and ψ. Now if $k_m(\zeta,\psi)$ is even, the last transition was from near wake to far wake. Since $|\vec{q}_s| <$ DM implies another section of far wake, there is no transition. However, if $k_m(\zeta,\psi)$ is odd, the last transition was from far wake to near wake. In this case, $|\vec{q}_s| <$ DM implies a section of far wake and a transition from near wake to far wake. Therefore, $k_m(\zeta,\psi)$ is incremented by one and using k equal to this new value of $k_m(\zeta,\psi)$:

$$\delta_T(\zeta,\psi,k) = \delta_\ell - \delta \tag{4.37}$$

Now consider the case $|\vec{q}_s| >$ DM, which implies a near wake segment. In this case, odd $k_m(\zeta,\psi)$ implies no transition. For even $k_m(\zeta,\psi)$ there is a transition from far wake to near wake. Therefore, $k_m(\zeta,\psi)$ is incremented by one and using k equal to this new value of $k_m(\zeta,\psi)$:

$$\delta_T(\zeta,\psi,k) = \delta_\ell - \delta + \Delta\psi \tag{4.38}$$

The transition age $\delta_T(\zeta,\psi,k)$ is $\Delta\psi$ larger than the relative age in this case. This is necessary because the trailing-wake segments identified by δ_ℓ actually extend from δ_ℓ to $\delta_\ell + \Delta\psi$.

In WG-71, the vectors from point P_η to the various points P_ℓ ($\vec{a}$ or $\vec{b}$ as appropriate) are only computed for the current value of δ_ℓ. Thus, the δ_ℓ loop always starts with an initial δ_ℓ for which the vectors from point P_η to points P_ℓ are computed but no induced velocities are computed. These vectors become the $\vec{b}$ for the trailing-wake segments whose induced velocity contribution is computed at the second δ_ℓ. At the other end of the near-wake segment (Eq. 4.37), the extra $\Delta\psi$ is not needed because this is the last δ_ℓ for which the induced velocity contribution is computed.

The two-tip vortex-line segments adjacent to P_η are always part of the near wake even if their $|\vec{q}_S|$ is less than DM. These are the two vortex-line segments which are combined into one curved vortex-line segment (Subsection 4.1.2). The relative ages δ_R of these segments are zero and $\Delta\psi$. Thus, unless the segments at relative ages δ_R of $-\Delta\psi$ and $2\Delta\psi$ are also part of the near wake, there will be a far wake to near wake transition (odd k) for $\zeta = 0$ and all ψ at $\delta_T(0,\psi,k) = -\Delta\psi$ and a near wake to far wake transition (even k) for $\zeta = 0$ and all ψ at $\delta_T(0,\psi,k) = \Delta\psi$.

Now consider $\vec{q}_F(\psi,\delta)$. At the start of a distortion iteration $\vec{q}_F(\psi,\Delta\psi)$ is set equal to zero. As the computation proceeds for $\delta = \delta_M = \Delta\psi$ the appropriate $\vec{q}_S$ are added to $\vec{q}_F(\psi,\Delta\psi)$. Next, δ_M is incremented to $2\Delta\psi$ and eventually δ will equal $2\Delta\psi$. At this point, an initial value for $\vec{q}_F(\psi, 2\Delta\psi)$ is needed. Now, all of the far wake elements which originally contributed to $\vec{q}_F(\psi,\Delta\psi)$ are $\Delta\psi$ older in age δ_ℓ, because the rotor has moved forward $\Delta\psi$. However, the relative age δ_R of these wake elements has not changed. The basic assumption of the near/far wake scheme is that changes in the induced velocity contribution of far wake elements are neglected whether due to changes in age δ_ℓ, or due to an improved knowledge of the distortion. Thus, $\vec{q}_F(\psi, 2\Delta\psi)$ is equal to $\vec{q}_F(\psi,\Delta\psi)$ plus the $\vec{q}_S$ of any new far-wake elements generated as the rotor moved forward by $\Delta\psi$. This can be generalized to say that the initial value of any $\vec{q}_F(\psi,\delta)$ is the final value of $\vec{q}_F(\psi,\delta - \Delta\psi)$.

Now consider case (2), updating. In case (1), the transitions $\delta_T(\zeta,\psi,k)$ and the far-wake induced velocities $\vec{q}_F(\psi,\delta)$ were determined. In case (2), this information will be used. The use of $\vec{q}_F(\psi,\delta)$ is clear. During any update $\vec{q}_F(\psi,\delta)$ is added to $\Sigma\vec{q}$, representing the contribution of the far-wake elements, which are otherwise neglected by definition.

The use of $\delta_T(\zeta,\psi,k)$ is to determine the ages δ_ℓ for which the induced velocity contribution will be updated (i.e., the near wake). Normally, δ_ℓ varies from $\delta + 2\pi M$ to zero in steps of $\Delta\psi$ (Fig. 25b). For this case, δ_ℓ varies in steps of $\Delta\psi$ from $\delta + \delta_T(\zeta,\psi,k)$ to $\delta + \delta_T(\zeta,\psi,k+1)$ for all odd k from 1 up to either $k_m(\zeta,\psi)$ or $k_m(\zeta,\psi) - 1$ with the exception that any $\delta_\ell < 0$ are neglected. This is because they represent wake elements which do not exist at the current value of δ. Thus, the δ_ℓ values will come in several small groups instead of one big group.

This completes the description of the near/far wake scheme applied to the basic distortion iteration. Its application to WG-71 is very similar. The primary difference is that WG-71 does not employ complete updating all of the time . Thus, the criteria $\delta < \delta_M$ is no longer sufficient to indicate updating (see Subsections 4.2.2 and 4.2.3). Another difference is that, when $\delta_M = \Delta\psi$, instead of using an odd or even $k_m(\zeta,\psi)$ to determine the type of transition which last occurred, an indicator MGO is used. This is explained in the description of the actual distortion iteration (Subsection 4.2.5).

4.2.2 Boundary Updating

The basic distortion iteration (Fig. 25) completely updates the computed $\vec{D}(\psi,\delta)$ each time the boundary between the computed and the assumed wakes (δ_M) is incremented by $\Delta\psi$. This complete updating is very expensive. Therefore, consider the other extreme of no updating at all until the end of a complete iteration. This approach has been tried [18] and it proved to be unsuccessful. However, it is useful to start with no updating and then add more and more updating until an accurate and efficient distortion computation is achieved. This subsection will consider boundary updating while the next subsection will consider general updating, both of which are used in WG-71.

Consider what no updating implies. As mentioned in Subsection 4.1.5, it implies that the δ loop (Fig. 25) is eliminated and that δ is always equal to δ_M. In addition to this, the δ_ℓ loop will no longer always run from $\delta + 2\pi M$ to zero. For the first step ($\delta_M = \Delta\psi$), the δ_ℓ loop will indeed run from $\Delta\psi + 2\pi M$ to zero. However, for the next step ($\delta_M = 2\Delta\psi$), the δ_ℓ loop will only run from $\Delta\psi$ to zero. This is because the contribution to the induced velocity of all of the wake having age δ_ℓ between $2\Delta\psi + 2\pi M$ and $\Delta\psi$ has already been computed. Thus only the contribution of the new wake generated when the rotor was indexed from $\delta_M = \Delta\psi$ to $\delta_M = 2\Delta\psi$ needs to be computed.

This may be confusing. In the first step the contribution of the wake between δ_ℓ equal to $\Delta\psi + 2\pi M$ and δ_ℓ equal to zero was computed. In the next step this same wake is referred to as running from a δ_ℓ of $2\Delta\psi + 2\pi M$ to a δ_ℓ of $\Delta\psi$. This is because δ_ℓ is measured from the rotor and when the position of the rotor is advanced by $\Delta\psi$ the δ_ℓ of any particular wake element is also increased by $\Delta\psi$.

Another way to look at this is to consider only where the wake is in relation to point P_η and to ignore the position of the rotor. The relative age δ_R developed in Subsection 4.2.1 (Eq. 4.35) is the equivalent of δ_ℓ, but is measured from point P_η instead of from the rotor. Now in the first step the contribution of the wake running from a δ_R of $2\pi M$ to a δ_R of $\Delta\psi$ was computed. In the next step this wake is still labeled with the same values of δ_R and the contribution of the new wake running from a δ_R of $\Delta\psi$ to a δ_R of $2\Delta\psi$ is computed. Thus, it is seen that no updating includes the assumption that the contribution to the induced velocity at point P_η of a wake element having a fixed relative age δ_R does not change as the rotor advances, increasing both the age δ of point P_η and the age δ_ℓ of the wake element.

Thus under the no updating assumption, the contribution of the wake having any given relative age δ_R is computed only once in a particular iteration. This implies that for any step where δ_M (and hence δ) is incremented by $\Delta\psi$ (except for the first step), the contribution of only that wake having δ_ℓ between $\Delta\psi$ and zero and δ_R between $\delta - \Delta\psi$ and δ is computed. This wake consists of the trailing-wake segments extending from $\delta_\ell = 0$ to $\delta_\ell = \Delta\psi$, which are identified with $\delta_\ell = 0$, plus the shed wake at $\delta_\ell = \Delta\psi$. The contribution of the bound vorticity is also computed. The contribution of the rest of the wake is assumed to remain unchanged.

Now consider boundary updating. For this case, the age δ remains equal to the boundary age δ_M (hence, the name boundary updating). However, the assumption that the induced velocity at point P_η does not depend on the age δ of point P_η is relaxed. This boundary updating only occurs at intervals of $n_{DM}(\psi)$ steps $\Delta\psi$ in the boundary age δ_M, where $n_{DM}(\psi)$ is an input to WG-71. Since only the distortion at the boundary $\vec{D}(\psi,\delta_M)$ is affected by boundary updating, each azimuth angle ψ can be treated independently. This allows $n_{DM}(\psi)$ to vary with ψ so that the wake along the sides of the rotor can be updated more often than the rest of the wake. This is desirable because the wake at the sides tends to distort more rapidly than the rest of the wake, due to the close proximity of the tip vortices from the various blades.

A distortion computation with only boundary updating is just like a computation with no updating until δ_M reaches a value such that:

$$n_{DM}(\psi) = (\delta_M - \Delta\psi)/\Delta\psi \tag{4.39}$$

Then, at all azimuth angles ψ for which Eq. 4.39 holds, the induced velocity contribution of the near wake is updated (recomputed) for age $\delta = \delta_M$. After this the computation proceeds with no more updating at these ψ until:

$$2\, n_{DM}(\psi) = (\delta_M - \Delta\psi)/\Delta\psi \tag{4.40}$$

Then the induced velocity contributions are updated again. This process continues with more updating at each integral multiple of $n_{DM}(\psi)$, until δ_M equals $2\pi m$ and an iteration is completed.

4.2.3 General Updating

General updating is the only place in the distortion computation used by WG-71 where the age δ is not required to be equal to the boundary age δ_M. Clearly, when δ is less than δ_M, the computed $\vec{D}(\psi,\delta)$ is an updated value. This is because there was another computation of this $\vec{D}(\psi,\delta)$ earlier in the iteration when δ_M was equal to this value of δ. Thus general updating could also be called distortion updating. Similarly, boundary updating could also be called induced velocity updating, since a boundary update at a given δ_M does not affect the distortion $\vec{D}(\psi,\delta)$ for ages δ less than δ_M, where the distortion has already been computed.

General updating occurs every ℓ_{DM} steps $\Delta\psi$ in the boundary age δ_M, where ℓ_{DM} is an input to WG-71. The distortion computation proceeds like a distortion computation with boundary updating only (Subsection 4.2.2) until δ_M reaches a value such that:

$$\ell_{DM} = (\delta_M - \Delta\psi)/\Delta\psi \tag{4.41}$$

At this point, a δ loop is activated, like the δ loop in Fig. 25a, where δ runs from $\Delta\psi$ to δ_M. This causes the distortion $\vec{D}(\psi,\delta)$ to be updated (recomputed) for all of these ages δ.

Since it would be wasteful to use a more expensive distortion computation technique for updating than for the original computation, the boundary updating scheme is applied to this distortion updating process. Thus, when recomputing $\vec{D}(\psi,\delta)$ for δ between $\Delta\psi$ and δ_M, the boundary updating computational scheme (Subsection 4.2.2) is used, but with δ_M replaced by $(\delta_D + \Delta\psi)$ in Eqs. 4.39 and 4.40. It will be recalled that δ_D is the boundary age between the most recently updated distortion and the older distortion. It is necessary to add $\Delta\psi$ to δ_D to make it analogous to δ_M, because δ_D is the age of the most recently completed distortion computation, while δ_M is the age for which the distortion is currently being computed, except during a general update.

The general updating interval ℓ_{DM} is not a function of azimuth angle ψ, since the distortion $\vec{D}(\psi,\delta)$ is being updated for all ages δ from $\Delta\psi$ to δ_M. The accuracy of the updated $\vec{D}(\psi,\delta)$ depends upon the $\vec{D}(\phi,\delta_\ell)$ for all azimuth angles ϕ, not just the ϕ which happens to be equal to ψ. Thus it is important to update the distortion at all azimuth angles ψ simultaneously.

A word about the choice of the updating intervals $n_{DM}(\psi)$ and ℓ_{DM}. First they should always be chosen such that some integer times $\Delta\psi$ times either $n_{DM}(\psi)$ or ℓ_{DM} will equal $2\pi m$. Thus, for $\Delta\psi = 15°$ and $m = 2$, the possible values of $n_{DM}(\psi)$ and ℓ_{DM} are 1, 2, 3, 4, 6, 8, 12, 16, 24, and 48. Also, ℓ_{DM} should be an integral multiple of all values of $n_{DM}(\psi)$. Typically, $n_{DM}(\psi)$ might have values of 3 and 6, depending on ψ, and ℓ_{DM} might be 12.

4.2.4 Averaging Incremental Distortion

For some cases the computed distortion shows a tendency to oscillate during the computation. The traditional cure for oscillations in an iterative computation is to average the results of successive iterations. This technique is used in WG-71. However, its application must be clearly thought out.

The obvious thing to do is to average the $\vec{D}(\psi,\delta)$ from two successive distortion iterations and use this average $\vec{D}(\psi,\delta)$ for the assumed distortion of the next iteration. This is not satisfactory, however, because general updating changes the value of $\vec{D}(\psi,\delta)$ during an iteration. General updating can be also thought of as another distortion iteration within the main iteration. Therefore, instead of averaging the distortion from two successive iterations, the incremental distortion is averaged over two successive computations.

This process is shown in the lower part of Fig. 26b. The old value of the distortion $\vec{D}_O(\psi)$ is the same as it was for the basic distortion iteration (Fig. 25, Subsection 4.1.5). Just before $\vec{D}_O(\psi)$ is set equal to $\vec{D}(\psi,\delta)$, the number stored in $\vec{D}_O(\psi)$ will be the old value of $\vec{D}(\psi,\delta - \Delta\psi)$, except when δ equals $\Delta\psi$. Thus the old value of the incremental distortion as age increases from $\delta - \Delta\psi$ to δ will be:

$$\Delta\vec{D}_o = \vec{D}(\psi,\delta) - \vec{D}_o(\psi) \tag{4.42}$$

The current value of the incremental distortion is $\Delta\psi\vec{q}_D(\psi)$, as before. The weighted average of these two values is used to increment the computed distortion:

$$\vec{D}(\psi,\delta) = \vec{D}(\psi,\delta-\Delta\psi) + w_f\,\Delta\psi\,\vec{q}_D(\psi) + (1-w_f)\,\Delta\vec{D}_o \tag{4.43}$$

where w_f is an input weighting factor, normally 0.5. For the special case $\delta = \Delta\psi$, the distortion $D(\psi,\delta - \Delta\psi)$ will always be zero. Thus

$$\Delta\vec{D}_o = \vec{D}_o(\psi) = \vec{D}(\psi,\delta) \tag{4.44}$$

and

$$\vec{D}(\psi,\delta) = w_f\,\Delta\psi\,\vec{q}_D(\psi) + (1-w_f)\,\vec{D}_o(\psi) \tag{4.45}$$

4.2.5 Actual Distortion Iteration

The actual distortion iteration used in WG-71 includes the various refinements described in Subsections 4.2.1 through 4.2.4. A simplified flow chart for the actual distortion iteration is given in Fig. 26. This subsection will amplify the flow chart without repeating all of the details already covered.

Consider the indicator MGO, which is used to indicate the current state of the distortion iteration. There are four possible values for MGO: 0, 1, 2, and 3. When MGO is either 0 or 1, the boundary age δ_M is equal to $\Delta\psi$. This is a special case because the contribution to the induced velocity of all of the wake between the ages δ_ℓ of $2\pi M + \Delta\psi$ and zero is being computed for the first time in the current iteration. The tests for $\delta_M = \Delta\psi$ shown in Fig. 26b are written as tests for MGO ≤ 1 in WG-71, because integer logic is faster. These values of MGO are also used to indicate the type of the last transition. For MGO = 0, the last transition was from far wake to near wake; i.e., $k_m(\zeta,\psi)$ odd. For MGO = 1, the last transition was from near wake to far wake; i.e., $k_m(\zeta,\psi)$ even. In WG-71 MGO is initially set equal to unity and $k_m(\zeta,\psi)$ is set equal to zero.

The cases MGO = 2 and MGO = 3 contain no information about transitions. Instead they indicate the use of general updating (Subsection 4.2.3). When MGO = 2, a general update is in progress. A boundary update is also in progress if $(\delta - \Delta\psi)/\Delta\psi$ is an integral multiple of $n_{DM}(\psi)$. When MGO = 3, the age δ is equal to the boundary age δ_M, and $\Delta\psi$ of new wake has been generated. If $(\delta_M - \Delta\psi)/\Delta\psi$ is an integral multiple of $n_{DM}(\psi)$, a boundary update will be in progress, but there is no general update as long as MGO = 3.

The flow chart in Fig. 26a consists of the δ_M and δ loops. The boundary age δ_M loop is the same as in Fig. 25a. However, the age δ loop is modified to provide for general updating. The loop runs from δ_1 to δ_M instead of from $\Delta\psi$ to δ_M. The initial age δ_1 is $\Delta\psi$ during a general update (MGO = 2) and δ_M, otherwise. The age δ_s of the latest general update is used to test for the start of the next general update (Subsection 4.2.3). The induced velocity contribution $\vec{q}_F(\psi,\delta)$ of the far wake is initialized to either zero (for $\delta = \Delta\psi$), or $\vec{q}_F(\psi,\delta - \Delta\psi)$. Finally, the test for the end of an iteration is made on δ_D instead of δ_M to conform with WG-71. However, δ_D will always be equal to δ_M at that point in the computation so the result is the same.

The flow chart in Fig. 26b consists of the azimuth angle ψ loop and the distortion computation. A new variable $\vec{q}_{TOT}(\psi)$ is introduced which was not in the basic distortion iteration (Fig. 25a). This $\vec{q}_{TOT}(\psi)$ contains the value of $\Sigma\vec{q}$ at azimuth angle ψ, from the previous step in the age δ loop. Thus, if $\Sigma\vec{q}$ was written as a function of ψ and δ, this $\vec{q}_{TOT}(\psi)$ would equal $\Sigma\vec{q}(\psi,\delta - \Delta\psi)$. Unless a boundary update is in progress, or $\delta_M = \Delta\psi$, the initial value of $\Sigma\vec{q}$ is:

$$\Sigma\vec{q}(\psi,\delta) = \vec{q}_{TOT}(\psi) = \Sigma\vec{q}(\psi,\delta-\Delta\psi) \tag{4.46}$$

since these induced velocity contributions are not recomputed (Subsection 4.2.2). The actual computation of the distortion $\vec{D}(\psi,\delta)$ is modified to allow incremental distortion averaging (Subsection 4.2.4).

The flow chart in Fig. 26c corresponds to the basic distortion iteration flow chart in Fig. 25b. The details of initializing δ_ℓ and finding the next transition are not repeated here. They can be found in Subsection 4.2.1, or the listing of WG-71 (Appendix I) can be consulted. The initial value of δ_ℓ is stored in δ_{ST} so that when $\delta_\ell = \delta_{ST}$, point P_ℓ is located, but no induced velocities are computed. The details of finding transitions and adding $\vec{q}_S$ to $\vec{q}_F(\psi,\delta)$ are not repeated here. These details can be found in Subsection 4.2.1 or in the listing of WG-71 (Appendix I).

4.3 Hovering Wake Geometry

Since hovering is a special case of forward flight, it might be expected that a forward flight-wake geometry computation for $\mu = 0$ would be satisfactory. This is not true, however. There is no relative wind in the hover case, so the wake does not move away from the rotor or from itself as fast as in the forward flight case. Thus, more wake must be included in the wake model which greatly increases the expense of the computation. There is a saving to compensate for this expense, however, since the hovering-wake geometry can be assumed to be axially symmetric. This eliminates the ψ-loop with a factor of $(2\pi/\Delta\psi)$ saving in computational expense.

Ideally, a special hovering-wake-geometry program, with an associated hovering airloads program, should be developed. These programs should use a polar coordinate system to take full advantage of the axial symmetry. However, to evaluate the usefulness of the techniques developed in WG-71 for the hovering case, an optimum program is not required. Thus, a simple modification of WG-71 for the hovering case has been developed. This modification makes use of axial symmetry without converting the entire program to polar coordinates (Subsection 4.3.1), and it provides specialized wake models for the hovering case (Subsection 4.3.2).

A corresponding airloads computation has not been developed. The axial symmetry assumption means that the circulation is the same for all azimuth angles. The details of the radial variation of bound circulation are not needed when the basic wake model is used (Subsections 2.1 and 2.8). Thus, the circulation γ_o required for the hovering wake geometry computation is provided by hand computations using momentum/blade element theory [58].

4.3.1 Axial Symmetry

Axial symmetry implies that the airloads and hence the circulation are the same at all azimuth angles ψ. It also implies that for any given radial station η and age δ the wake geometry at point P_η will be the same for all ψ. Thus, it is necessary to compute the airloads and the wake geometry at only one value of ψ. This reduces the amount of computation required by a factor of $(2\pi/\Delta\psi)$.

While the tip vortex distortion and hence, the tip vortex geometry is computed for only one value of ψ, it is used at all azimuthal stations ϕ when computing induced velocities. If the distortion were computed in polar coordinates (r,θ, z), this would be no problem, since the r,θ, and z components of the distortion would be the same for all ϕ. However, WG-71 is written in x, y, z coordinates and while the actual distortion vector does not change with ϕ, it's x and y components do change with ϕ.

To maximize the use of the existing program WG-71, the x, y, z coordinate system is retained. The normal table of $\vec{D}(\psi,\delta)$ is used and $\vec{D}(\psi,\delta)$ is computed using $\vec{q}_D(\psi)$ in the usual way (Fig. 26b). However, the main ψ-loop of Fig. 26b is executed for $\psi = 0$ only. This gives a value for $\vec{q}_D(0)$ only. To compute the rest of the $\vec{q}_D(\psi)$, define the x,y,z components of $\vec{q}_D(\psi)$:

$$\vec{q}_D(\psi) = q_{Dx}(\psi)\,\vec{i} + q_{Dy}(\psi)\,\vec{j} + q_{Dz}(\psi)\,\vec{k} \tag{4.47}$$

Now:

$$q_{Dx}(\psi) = q_{Dx}(0)\cos\psi + q_{Dy}(0)\sin\psi \tag{4.48}$$

$$q_{Dy}(\psi) = q_{Dy}(0)\cos\psi - q_{Dx}(0)\sin\psi \tag{4.49}$$

$$q_{Dz}(\psi) = q_{Dz}(0) \tag{4.50}$$

4.3.2 Hovering Wake Models

The most important difference between forward flight-wake models and hovering-wake models is that a lot more wake is included in hovering-wake models. A $\mu = 0.2$ distortion computation typically uses $m = 2$ and $M = 2$, while a hovering-distortion computation typically uses $m = 6$ and $M = 12$. Thus, the distortion is computed for the 6 turns (12π in age) of wake nearest the rotor, and the contribution to the induced velocity at point P_η of all wake elements within 12 turns (24π in age) of point P_η is computed. To reduce computation expense, $\Delta\psi$ is normally doubled, from 15° to 30°.

In WG-71, all of the wake greater than $2\pi M$ in age from point P_η is neglected. For the hovering case, this wake is represented by a semi-infinite vortex cylinder for each trailing vortex line (there is no shed wake in hovering due to the axial symmetry assumption). This cylinder is specified by point P_ℓ on its upper edge. The radius of the cylinder is (when normalized by R):

$$r = \ell + D_{\ell y}(0, \delta_\ell) \tag{4.51}$$

where

$$\delta_\ell = \delta + 2\pi M \tag{4.52}$$

Lamb [35] has shown that the velocity potential $\Delta\phi_P$ at point P due to a closed vortex element of circulation Γ is:

$$\Delta\phi_P = \frac{\Gamma}{4\pi}\,\omega \tag{4.53}$$

where ω is the solid angle subtended at P by the closed vortex element. Define ρ as the distance from the center of the vortex ring to point P (normalized by r). Let x be the radial distance from the axis of the vortex ring to point P (normalized by r) and z to be the distance from point P to the plane of the vortex ring (normalized by r). Now

$$\rho = \sqrt{x^2 + z^2} \tag{4.54}$$

To define the solid angle ω, consider a sphere of radius ρ centered at point P, and a conical surface connecting the vortex ring and point P. Let A be the area on the surface of the sphere which is surrounded by the intersection of the sphere and the conical surface. When

$$\omega = A/\rho^2 \tag{4.55}$$

Now, if ρ is so large that the curvature of the surface of the sphere in area A can be neglected, the area A can be approximated by a planar ellipse of minor axis $r(z/\rho)$ and major axis r. Normalizing A by r^2:

$$A \simeq \pi z/\rho \tag{4.56}$$

Therefore

$$\omega \simeq \pi z/\rho^3 \tag{4.57}$$

This leads to an approximation for the velocity potential

$$\Delta\phi_p \simeq \frac{\Gamma}{4}\,\frac{z}{(x^2+z^2)^{3/2}} \tag{4.58}$$

Now consider a semi-infinite vortex cylinder extending from z to ∞. This cylinder is made up of an infinite number of vortex rings, each having circulation $d\Gamma/dz$. Therefore, the velocity potential of this cylinder at point P is

$$\phi_p = \frac{1}{4}\frac{d\Gamma}{dz}\int_z^{\infty}\frac{z\,dz}{(x^2+z^2)^{3/2}} = \frac{1}{4}\frac{d\Gamma}{dz}\left(\frac{1}{\sqrt{x^2+z^2}}\right) \tag{4.59}$$

The x and z components of the induced velocity at point P are

$$q'_{Px} = \frac{1}{rR}\frac{\partial\phi_p}{\partial x} = \frac{-1}{4rR}\frac{d\Gamma}{dz}\frac{x}{(x^2+z^2)^{3/2}} \tag{4.60}$$

$$q'_{Pz} = \frac{-1}{rR}\frac{\partial\phi_p}{\partial z} = \frac{1}{4rR}\frac{d\Gamma}{dz}\frac{z}{(x^2+z^2)^{3/2}} \tag{4.61}$$

where the signs are chosen in accordance with the conventions of WG-71.

To calculate $d\Gamma/dz$, observe that circulation $n_b\Gamma$ (n_b = number of blades) will accumulate as the rotor turns 2π in azimuth. During the same time, the wake will move downwards at velocity $\lambda\Omega R$ an amount

$$rRz = \frac{2\pi}{\Omega}\lambda\Omega R \tag{4.62}$$

Therefore

$$\frac{1}{rR}\frac{d\Gamma}{dz} = \frac{n_b\Gamma}{2\pi\lambda R} \tag{4.63}$$

Now using

$$T = \frac{1}{2}n_b\rho\Omega R\Gamma R \tag{4.64}$$

$$C_T = \frac{T}{\rho\pi R^2(\Omega R)^2} \tag{4.65}$$

$$\lambda = \sqrt{C_T/2} \tag{4.66}$$

It can be shown that

$$\frac{1}{rR}\,\frac{d\Gamma}{dz} = 2\lambda\,\Omega R \tag{4.67}$$

Normalizing the x and z components of the induced velocity with the tip speed ΩR and using Eq. 4.67:

$$q_{Px} = -\frac{\lambda}{2}\,\frac{x}{(x^2+z^2)^{3/2}} \tag{4.68}$$

$$q_{Pz} = \frac{\lambda}{2}\,\frac{z}{(x^2+z^2)^{3/2}} \tag{4.69}$$

Now expressions for x and z are needed. The vortex rings are assumed to be parallel to the TPP. Define the x_p and y_p as the $\vec{i}$ and $\vec{j}$ components of a vector, parallel to the TPP, which goes from the axis of the vortex cylinder to point P (point P_η in the notation of WG-71).

$$x_p = D_{\eta x}(0,\delta) \tag{4.70}$$

$$y_p = \eta + D_{\eta y}(0,\delta) \tag{4.71}$$

The magnitude of this vector is rx.

$$x = \sqrt{x_p^2 + y_p^2}\,/r \tag{4.72}$$

Now z is the vertical (k) component of the vector from point P_η to point P_ℓ:

$$z = \frac{1}{r}\Big[\mu \tan i\,(\delta_\ell - \delta) - a_0(\ell - \eta) + D_{\ell z}(0,\delta_\ell) - D_{\eta z}(0,\delta) \tag{4.73}$$

where δ_ℓ is given by Eq. 4.52.

Finally, the $\vec{i},\vec{j},\vec{k}$(x,y,z) components of the induced velocity contribution $\vec{q}$ of the vortex cylinder are

$$q_x = q_{px}\left(x_p \Big/ \sqrt{x_p^2 + y_p^2}\right) \tag{4.74}$$

$$q_y = q_{px}\left(y_p \Big/ \sqrt{x_p^2 + y_p^2}\right) \tag{4.75}$$

$$q_z = q_{pz} \tag{4.76}$$

The hovering-wake geometry computation computes the tip vortex distortion only. Therefore, the geometry of the inboard trailing wake is normally determined by the rigid wake assumption (Subsection 1.5.1). However, Landgrebe [23] has developed an empirical wake geometry correlation for both the tip vortex and the inboard trailing wake. This correlation shows that the inboard trailing wake is a vortex sheet whose vertical velocity is much smaller near the root than near the tip. A set of formulas are given in [23] for the geometry of this inboard vortex sheet. These formulas have been included as an option in the hovering wake geometry computation when an inboard trailing-vortex sheet model is used.

SECTION 5

RESULTS AND CONCLUSIONS

The results presented here consist of top and side views of the tip vortex geometry (for example, Fig. 1) and of plots of airloads (lb/in) vs. azimuth angle ψ at a given radial station η (for example, Fig. 2). These results are compared with measured data from experiments when such data are available. The calculated tip vortex geometry is also compared with the classical rigid wake tip vortex geometry (see Subsection 1.5.1). Unfortunately, there is no case where experimental data is available for both tip vortex geometry and airloads.

The results are primarily designed to show the effects of the various models and of computational parameters on both tip vortex geometry and airloads. The results presented here are restricted to those obtained using the final versions of the airloads program LDS-73 and the wake geometry program WG-71 as listed in Appendices F and I. Thus, anyone using these programs should be able to duplicate these results. These programs and their predecessors have been continuously developed and modified over the past ten years. It was not financially practical to repeat all of the cases previously run, using these older programs. Furthermore, some wake models which were evaluated on older programs have been eliminated from the current programs to save computer memory space (core storage). Thus, these older results will be mentioned occasionally in cases where the modifications to the programs involved do not significantly affect the point in question.

Both airloads and tip vortex geometry have been computed at various times for one, two, three, four, and six bladed rotors at various advance ratios from zero to 0.45. The results presented here are restricted to four- and six-bladed rotor airloads, with the corresponding tip vortex geometries, plus a one-blade tip vortex geometry case (for comparison with smoke tests). These results are representative of all of those obtained at low to moderate advance ratios (μ = 0.1 to 0.3). Due to the neglect of torsional motion in the blade motion solution the airloads program is not suitable for high advance ratios. Hover (μ= 0) is treated as a special case (see Subsection 4.3).

Most of the results presented are for the four-blade case with advance ratio μ = 0.18. This has been the standard airloads case for many years and a large background of computational experience has been developed. A standard set of inputs (i.e., wake model plus computational parameters) has been developed for this case, which is designed to give good agreement with the measured data at reasonable expense. The effects of changing the various input parameters are demonstrated by permutations from this standard set of inputs. This same standard set of inputs has been tested on other cases (such as the six-blade case) to ensure that it is not too specialized.

The largest and most important airloads occur on the outer portion of the rotor. Thus, the radial stations chosen are more closely spaced outboard and the results presented are normally for the outboard radial stations. It is not necessary to present the airloads results from all six radial stations for each permutation of the inputs. Instead the airloads results are normally presented for only one of three outboard radial stations, unless there is a significant difference between the results at the various radial stations. The radial station chosen is varied so that examples of results from all of the outboard radial stations are shown. Results from inboard radial stations are shown only when they are of special interest.

The tip vortex geometry plots show the geometry of the tip vortex from one blade at the instant when that blade is at azimuth angle ψ_s. The geometry will be different for each different ψ_s, because the age δ of the various points P_η in the wake is changed by changing ψ_s. However, the changes with increasing age are normally slow enough so that results at one or two values of ψ_s (normally ψ_s = 180° or 0°) are enough to show the wake behavior.

5.1 Wake Model Parameters

This subsection considers the effects of variations in the wake model on the results. Only those wake model parameters which apply to both wake geometry and airloads computations will be considered. More specialized wake model parameters will be considered later (Subsections 5.2 and 5.3). The results presented are mostly rigid wake airloads for the four-blade, $\mu = 0.18$ case (Table 2). However, these wake model parameters have also been tested for various other cases for both airloads and wake geometry, using the older versions of the programs LDS-73 and WG-71.

Consider $\Delta\phi$, the azimuthal extent of the straight trailing vortex line and vortex sheet segments used in the wake model. Rigid wake airloads were computed for $\Delta\psi$ = 15° with $\Delta\psi$ = 15°, 30°, and 45°. The results at $\eta = 0.85$ are typical and Fig. 27 shows those for $\Delta\phi$ = 15° and 30°, compared with measured flight test data [49]. The results for $\Delta\phi$ = 45° are not presented, but they show even larger differences from the $\Delta\phi$ = 15° results. The current programs LDS-73 and WG-71 cannot be used with $\Delta\phi$ less than 15°. However, older versions of both programs have been run with $\Delta\phi$ = 7.5°. The results were very similar to those with $\Delta\phi$ = 15°, for rigid wake cases. For distorted wake cases, some extra bumps in the tip vortex were picked up but the overall pattern remained the same. Thus, the distorted wake airloads for $\Delta\phi$ = 7.5° were very similar to those for $\Delta\phi$ = 15°.

It is concluded that $\Delta\phi$ = 15° is satisfactory for rigid wake airloads computations and for the computation of the gross tip vortex distortion. For the details of the tip vortex distortion very near a rotor blade, or another tip vortex, very small $\Delta\phi$, much less than one degree, would be required. This is clearly impractical for an overall computation of the entire wake, as considered here. Any computations of local tip vortex distortion should be done as a seperate problem isolated from the computation of the rest of the wake. For cases where the expense of computation must be reduced even at some cost in accuracy, a rigid wake computation with $\Delta\phi = 30°$ is suggested (see Fig. 27). However, $\Delta\psi = 15°$ should be retained, or peaks in the airloads may be lost.

The azimuthal extent of the wake included in airloads computations is $2\pi m$. Normally, m is 4 at $\mu = 0.10$, 3 at $\mu = 0.15$, and 2 at $\mu = 0.20$. To test the effect of m, rigid wake airloads were computed for a four-blade rotor at $\mu = 0.18$ with m = 1, 2, and 3. The results for m = 1 and m = 2 are shown in Fig. 28 at $\eta = 0.95$. The results for m = 3 coincide with the m = 2 curve. A set of distorted tip vortex airloads for m = 2 and m = 3 was also computed for this case. Again, the results were practically identical. Runs made with the older versions of LDS-73 and WG-71 with M = m for various advance ratios confirm these results. Thus, $m = 0.4/\mu$ is the suggested criterion for m, with hover (μ=0) being a special case (Subsection 4.3.2).

Now consider the inboard trailing wake. Rigid wake airloads were computed with, and without, the inboard trailing wake for the four-blade, $\mu = 0.18$ case. At the outer two radial stations there is some difference, while at the 75% radius (Fig. 6) and inboard there is a marked difference, especially at $\psi = 0$. Wake geometry computations made using the older version of WG-71 confirm this. Thus, the inboard trailing wake should not be neglected. This leaves the question of how to model it. The n-trailing-vortex-line model (Subsection 2.2) is not available in the current LDS-73; however, runs made for n = 6, 10, and 20 with an older version of LDS-73 showed very similar results at outboard radial stations (η >.7) when compared with the basic wake model (Subsection 2.1). A similar comparison was made with an older version of WG-71 with similar results. This comparison has not been repeated using the current WG-71 due to the great expense involved.

The basic wake model has provision for either a large-core vortex line model or a vortex sheet model of the inboard trailing wake (Subsection 2.6). Rigid-wake airloads were computed using these two options for a four-blade, $\mu = 0.18$ case. The results at the outer three radial stations are very similar, but from $\eta = 0.55$ (Fig. 29) inboard the large-core vortex line model shows better agreement with the measured flight test data [49] near $\psi = 0$. This is surprising, since the vortex sheet model was expected to be closer to reality.

To check this result, rigid-wake airloads were computed for the six-blade, $\mu = 0.108$ case (Table 2) using both models. There is no measured data at $\eta = 0.55$, but the $\eta = 0.40$ station shows the same behavior near $\psi = 0$ as the four-blade case, with the vortex sheet model predicting very large airloads while the large-core vortex line model agrees with the measured data [51]. A tip vortex geometry computation was also run, using the vortex sheet model, for the four-blade, $\mu = 0.18$ case. The results for $\psi_s = 0°$ (Fig. 30b) are compared with the large-core vortex line model results (Fig. 30a). The large bump in vortex sheet model geometry near $\psi = 180°$ can be traced to the influence of the vortex sheet inboard trailing wake.

To test the sensitivity of these results to the vortex sheet vortex core radius ρ_{vs}, values of ρ_{vs} as large as 0.050 (compared to the normal value of 0.0025) were tried for the four blade, $\mu = 0.18$ case. The details of the airloads results changed but the error near $\psi = 0°$ at the inboard radial stations remained. Thus, it is concluded that the inboard trailing wake partially rolls up so that its circulation distribution is more nearly circular than planar. This is not to say that it rolls up into a tight, concentrated vortex similar to the tip vortex, because the use of a small-core vortex line model leads to large airloads peaks at the inboard radial stations which do not appear in the measured data.

Now consider the shed wake. Rigid wake airloads were computed for the four-blade, $\mu = 0.18$ case: with no shed wake, with near shed wake only, and with both near- and far-shed wake. The results show that the far-shed wake has almost no effect for this case and that the near-shed wake effect is small. The tip vortex distortion was also computed for this case with and without shed wake (there is no near-shed wake in a distortion computation). The distortion results were not changed by the shed wake. Finally, rigid-wake airloads were computed for the six-blade, $\mu = 0.108$ case (Table 2), with and without shed wake. The shed-wake effect on the results is larger for this six-blade case than for the four-blade case. These six-blade airloads results are plotted at the 85% radius in Fig. 31, including some measured flight-test data [51].

These results were obtained using the large-core vortex-line far-shed-wake model and the vortex sheet near-shed-wake model (Subsection 2.7). Previous results from older versions of LDS-73 and WG-71, which used a small core vortex line shed-wake-model, showed much larger shed-wake effects. It is now thought that these larger effects were due to spurious peaks in the induced velocity due to the use of the small-core vortex-line shed-wake-model. Thus it is concluded that the far-shed wake can be safely neglected, which will save some computational expense.

The near-shed wake is another matter. To understand the larger near-shed-wake effects in the six-blade case compared to the four-blade case, consider their reduced frequencies:

$$k = \frac{\omega b}{V} = \frac{\omega b}{\Omega R} \tag{5.1}$$

where ω is the predominant frequency of the harmonic airloads. The semi-chord to radius ratio (b/R) is about the same for both cases (.025 and .030), but the frequency ratio of the harmonic airloads (ω/Ω) is not. In the four-blade case, the first, second, and third harmonics predominate. Taking the second harmonic as typical, the reduced frequency k becomes 0.050. In the six-blade case, the first, second, third, fifth, and seventh harmonics are all important. Taking the third harmonic as typical, k becomes 0.090. Miller [9] has developed an approximate expression for the two-dimensional lift deficiency function F when k is less than 0.5 which agrees well with Theodorsen:

$$F = \frac{1}{1 + k(\pi/2)} \tag{5.2}$$

This gives values of F of 0.93 and 0.875 for the four-blade and the six-blade cases, respectively. The shed-wake effect shown in Fig. 31 seems to agree with this estimate of F.

5.2 Wake Geometry Results

5.2.1 Computational Parameters

The computational parameters are the near/far wake parameter DM and the updating parameters ℓ_{DM} and $n_{DM}(\psi)$, see Subsection 4.2. The values for these parameters were determined by making a series of runs with decreasing values of the parameters until further decreases no longer had a significant affect on the results. A total of fourteen runs were made, using an older version of WG-71, for the one blade, $\mu = 0.10$ case (Table 2). The results of this study were that DM should be less than ten percent of the induced downwash ($\lambda -$ tan i) and that $\ell_{DM} = 12$ together with $n_{DM}(\psi) = 3$ along the sides of the rotor and $n_{DM}(\psi) = 6$ elsewhere are satisfactory.

To make sure that these results are still valid for the current version of WG-71 several runs were made for the four blade, $\mu = 0.18$ case. For the standard set of inputs DM = 0.0020, $\ell_{DM} = 12$, $n_{DM}(\psi) = 3$ (from $\psi = 45°$ to $135°$ and from $\psi = 240°$ to $330°$)

and $n_{DM}(\psi) = 6$ elsewhere. The tip vortex geometry obtained using these inputs is shown in Fig. 1 ($\psi_s = 180°$) and Fig. 30a ($\psi_s = 0°$). Runs were made for DM = 0.0010 and DM = 0.0005 together with the standard values of ℓ_{DM} and $n_{DM}(\psi)$. The results were essentially the same as those for DM = 0.0020. The DM = 0.0005 results are shown in Fig. 32 for $\psi_s = 180°$. Another run was made with DM = 0.0020, $\ell_{DM} = 4$, and $n_{DM}(\psi) = 2$ (for all ψ). These results are shown in Fig. 33 for $\psi_s = 180°$. Comparing Fig. 33 with Fig. 1, there are some small differences; however, they are in areas far from the rotor and hence are not important in airloads computations. Thus, it is concluded that the standard values of DM, ℓ_{DM}, and $n_{DM}(\psi)$ are small enough to give satisfactory results.

A comparison of the execution times required for WG-71 using various values of DM, ℓ_{DM}, and $n_{DM}(\psi)$ is given in Table 3. The execution time for the corresponding airloads computation using LDS-73 is also given. While the exact times are very dependent on the computer used (see Table 3), the relationship between the airloads computation and the wake geometry computation is not. It is estimated that a four-blade, $\mu = 0.18$ case with DM = 0.002, $\ell_{DM} = 1$, and $n_{DM}(\psi) = 1$ (i.e., complete updating) would require 400 to 500 seconds. A similar case without the near/far wake approximation (i.e., DM = 0.0) might require 5000 to 10000 seconds.

5.2.2 Comparison with Smoke Data

Qualitative experimental data on the tip-vortex geometry in forward flight is available from smoke tests and water tunnel tests. However, there is a lack of accurate quantitative data. Thus, the accuracy of the wake-geometry computation can only be evaluated qualitatively by comparison with these smoke tests or indirectly through airloads computations. The indirect technique does not work because of the current poor understanding of close blade-tip vortex interactions. Thus, we are left with a qualitative evaluation.

The clearest smoke pictures available are the culmination of several years of part-time work by John Thompson, F.S. Stoddard, M.W. Hurst, and Gregg Erickson. The experimental apparatus and technique are described by Erickson [59]. The results used here were taken by Erickson after his 16.62 project [59] was completed. The case chosen is a one-blade case at $\mu = 0.10$. Both two- and four-blade cases were also run at advance ratios up to 0.3; however, the one-blade photographs are the clearest. This is because the wake of the extra blades and the blades themselves interact with the visualized tip vortex (i.e., the one with smoke injection), causing the smoke to disperse in places.

Photographs of the smoke were taken simultaneously from the side and from the bottom. An attempt was made to reduce these simultaneous photographs to quantitative wake-geometry data. However, the reduced results did not make sense. The length of the rotor blade came out wrong, for example, and extra bumps appeared in the reduced data which did not appear in the photographs. It was concluded that the location of the side camera might have been in error, which would explain the problem.

The tip vortex geometry was computed for this one blade, $\mu = 0.10$ case (Table 2). The results are compared with some partially reduced smoke data in Fig. 34a ($\psi_s = 0°$) and Fig. 34b ($\psi_s = 180°$). Only the side view is shown and to clarify the presentation, the far side of the wake ($\psi = 0°$ to $180°$) is dashed. It is believed that the lower location of the far side of the wake in the smoke data compared to the computed data is partially due to an error in the vertical location of the side camera. Otherwise there is good qualitative agreement. Similar comparisons have been made for both two- and four-blade cases using an older version of WG-71 and the qualitative agreement is also good, although there are some gaps in the smoke data due to dispersion of the smoke as described above.

The side photograph corresponding to Fig. 34b is shown in Fig. 35. The corresponding bottom photograph is not shown because it was taken through a fish-eye lens which caused confusing distortion. The complete computed tip vortex geometry corresponding to Figs. 34a and 34b is shown in Fig. 36a ($\psi_s = 180°$) and Fig. 36b ($\psi_s = 0°$).

5.2.3 Other Forward Flight Results

All of the wake-geometry results presented here are computed using circulation distributions computed by LDS-73 for a rigid wake. To test the validity of this, a wake geometry computation was made for the four-blade, $\mu = 0.18$ case using a circulation distribution computed by LDS-73 for the tip vortex distortion shown in Figs. 1 and 30a. The results were identical with those in Figs. 1 and 30a. For a comparison of the airloads associated with these two circulation distributions (i.e., rigid wake and distorted wake) see Fig. 2.

As an example of the computed tip vortex geometry of a different case, the six-blade, $\mu = 0.108$ case was run. The results are presented in Figs.37a ($\psi_s = 180°$) and 37b ($\psi_s = 0°$). The tip vortex geometry far downstream (y greater than 3.5) may not be very accurate since the location of much of the wake which contributes to the induced velocity in that region is extrapolated. However, this is not important, since that part of the wake is too far away from the rotor to significantly affect airloads. A comparison of rigid vs. distorted wake airloads for this case will be shown in Subsection 5.3.

5.2.4 Discussion of Forward Flight Results

The most important result of these tip-vortex-geometry computations is that, over the forward portion of the rotor, the tip vortex tends to stay in the tip-path plane until the tip vortex from the next blade comes by to push it down (see Figs. 1 and 37a). This result has important consequences in airloads computations (Subsection 5.3). This subsection will consider the physics behind this important result.

The circulation of the tip vortex is such that it induces downwash on its concave side and upwash on its convex side when viewed from the top. Consider the tip vortex of the $\psi = 180°$ blade of a four-blade rotor at $\mu = 0.18$ (Fig. 1). Between $\psi = 180°$ and $\psi = 105°$, this tip vortex is the outermost in a nest of tip vortices from the various blades. Thus, it experiences an induced upwash from these other tip vortices. It will also experience an induced downwash from the inboard trailing wakes of these blades, but this is smaller because the wake which induces it is father away. Thus, there is a net induced upwash on the tip vortex in this region. At about $\psi = 105°$ on the tip vortex, the $\psi = 90°$ blade comes by (the vertical line from the hub in Fig. 1). The tip vortex from this $\psi = 90°$ blade is outboard of the tip vortex from the $\psi = 180°$ blade and hence induces a large downwash. However, a new source of upwash delays the effect of this downwash 15°, as discussed below.

There is a similar situation in the six-blade, $\mu = 0.108$ case (Fig. 37a). In this case, the next blade is at $\psi = 120°$ instead of at $\psi = 90°$ and its tip vortex pushes the tip vortex of the $\psi = 180°$ blade down at $\psi = 120°$ and at $\psi = 105°$. However, observe that the returning part of the tip vortex of the $\psi = 180°$ blade is inducing upwash on itself at $\psi = 90°$. This, together with upwash induced by similar returning tip vortices from the other blades, causes the tip vortex of the $\psi = 180°$ blade to go back up at $\psi = 90°$. A similar, but less intense,effect is also present in the four-blade, $\mu = 0.18$ case, causing the $\psi = 90°$ point to remain in the tip-path plane.

In summary, there is a sound physical explanation for the behavior of the tip vortex over the forward portion of the rotor, which clearly indicates that the tip vortex should be closer to the tip-path plane than the rigid-wake model would suggest.

5.2.5 Hovering Results

The hovering wake geometry program has not been modified to include the latest modifications to WG-71. Thus, only old results from 1971 and 1972 are available. The last runs made were for comparison with some smoke-test data by Landgrebe [23]. The results are shown in Fig. 38. The computed geometry does not contract rapidly enough near the rotor and hence does not go down fast enough either. This is true of Landgrebe's computations [23] also. It is thought that this discrepancy is due to a lack of understanding of the details of the formation and roll-up of the tip vortex very near the blade. The vortex sheet plus the vortex line (VSUL) tip-vortex model (Subsection 2.5.3) was not used in these computations; however, it is probably too crude to solve this problem in any case.

When the hovering wake-geometry computation was originally developed, the results were unstable. The technique of averaging the two most recent computations of the incremental distortion (Subsection 4.2.4) was developed to solve this problem. Since then, Landgrebe [23] and Widnall [24] have demonstrated both experimentally and theoretically that the wake beyond the first 2π or so, probably is unstable. Thus, the geometry computed in this region represents an average value of the unstable oscillations.

5.3 Airloads Results

The original intention was to compute airloads both with a rigid wake geometry and with a distorted wake geometry and then to compare the results. However, very large peaks appeared in the distorted wake airloads which either did not exist at all in the measured airloads or were much smaller. These peaks can easily be traced to the location of the tip vortex much closer to the rotor, over the forward portion of the rotor, in the distorted wake case (see Figs. 1 and 37a, for examples). This closer location of the tip vortex is explained physically in Subsection 5.2.4. There remains the task of explaining the lack of these large peaks in the measured airloads.

The first thought was the lack of a tip vortex core. A solid body rotation vortex core model was introduced and various values of the vortex core radius ρ_c (normalized by rotor radius R) were tried. It was found that $\rho_c = 0.05$ (about one chord for the four-blade case) reduced the peaks to about the right size. This made the distorted wake airloads agree with the measured airloads about as well as the rigid wake airloads. Unfortunately, measurements of tip vortex characteristics show that ρ_c is more than an order of magnitude smaller than this (Subsection 2.4). The current best estimate of ρ_c for the four-blade case is 0.0025, which results in excessively large peaks in the distorted wake airloads. Thus, it became necessary to search for other close blade-vortex interaction effects to explain the lack of these peaks in the measured airloads.

The models developed for the various close blade-vortex interaction effects are described in Subsections 2.4, 2.5, and 3.2.4. These effects are evaluated here by comparing the results of a standard case, which includes all of these effects, with the results of removing the various effects one at a time. The effects of changing various parameters, such as the vortex-core size, are evaluated in a similar fashion. The four-blade, $\bar{\mu} = 0.18$ case (Table 2) is used for these evaluations. The standard set of inputs, about which perturbations are made is shown in Table 4.

5.3.1 Lifting-Surface Solution

A close blade-vortex interaction induces a rapid spanwise variation of bound circulation on the blade. This, in turn, generates a trailing wake which induces airloads of the opposite sense on the blade (see Subsection 2.5.2). Originally, this extra trailing wake was represented by an approximate lifting-line solution. However, Johnson [52,53] developed a lifting-surface solution for this case. An approximate, incompressible version of this solution is used here (Subsection 3.2.4).

The rigid wake (Fig. 39) and distorted wake (Fig. 40) airloads at 85% radius are shown. The results at other outboard radii are even more dramatic. In these results, the old lifting-line approximation is not used when the lifting-surface solution is removed. Instead, the effects of the extra trailing wake are simply neglected. This is because the lifting-line approximation is no longer available in the LDS-73 program. However, experience with older versions of LDS-73 indicates that the lifting-line approximation reduces the peaks by about half and the lifting-surface solution by half again. Thus, this extra trailing wake must be represented in airloads computations and the lifting-surface solution is by far the best way to do this.

5.3.2 Tip Vortex-Core Radius

The effect of varying the burst vortex-core radius is shown in Fig. 41, at the 85% radius. The two values of ρ_g shown, bracket the standard value of 0.10R. For distorted wake airloads at 85% radius, with $\rho_g = 0.10R$, see the solid line in Fig. 40. These results at 85% radius are typical of the outboard radial stations. Unfortunately, there is no sound theoretical or experimental basis for accurately estimating ρ_g at present. The results shown in Fig. 41 indicate that this problem should be carefully studied so that an accurate estimate of ρ_g can be obtained.

Distorted wake airloads were computed for the initial vortex core radius ρ_k equal to: .0050R, .0015R, .0010R, and .0005R as well as the standard value .0025R. The results for all except the smallest value were almost equal. The results for $\rho_k = .0005R$ are compared to the standard case in Fig. 42, at 95% radius. This radius is typical and the differences shown are not very significant. It should be realized, however, that this comparison would be dramatically different if tip vortex-core bursting were not used here. A hint of this is seen in the effect of variations in ρ_g (Fig. 41). To summarize, uncertainties in the initial vortex radius ρ_k are not important as long as tip vortex-core bursting is used, but a more accurate estimate of the burst vortex-core radius ρ_g would be useful.

5.3.3 Tip Vortex-Core Bursting

Tip vortex-core bursting is described in Subsections 2.5.1 and 3.2.4. The effect of burst vortex-core radius ρ_g was considered in Subsection 5.3.2. This subsection will consider the other parameters associated with bursting.

First, consider propagation of tip vortex-core bursting (Fig. 43). The important effect is at the 75% radius,as shown. At the outer radial stations there is not much effect. This is not very surprising because the close blade-tip-vortex interactions which occur near the blade tip are the trigger which initiates bursting and hence propagation. Thus, propagation does not have time to build up at these outboard stations. However, the effect is important at the 75% radius and nothing else has been found which can replace it.

To show the effect of tip vortex-core bursting alone, without propagation, this is compared with the no-bursting case in Fig. 44a (95% radius), Fig. 44b (85% radius), and in Fig. 44c (75% radius). The use of bursting without propagation is seen to make very little difference, except for an extra secondary peak at about $\psi = 90°$. This peak is caused by bursting of the tip vortex from the $\zeta = 180°$ blade (i.e., the blade 180° away from the blade upon which airloads are computed). Since this secondary peak does not appear in the measured airloads, it is concluded that the very simple model of vortex bursting used here (i.e., an increase in tip vortex-core size from .0025R to

.10R) is inadequate. However, it is not possible to construct a better model without a better understanding of the whole process of tip vortex-core bursting, which would require a complete theoretical and experimental study of the subject. It is recommended that such a study be undertaken. Until then, the current model will be used.

One criterion for tip vortex-core bursting is that the tip vortex pass within (FVB)ρ_c of the blade, where FVB is an input constant and ρ_c is the tip vortex-core radius. The effect of changing FVB from the standard value of 20.0 to 1.0 is shown in Fig. 45, at 95% radius. The value of FVB = 1.0 is probably much more realistic; however, 20.0 is made standard to correspond to the standard rigid wake case where the tip vortex is unrealistically far away from the rotor. In any case, the airloads results are clearly not very sensitive to FVB for the distorted wake case.

The other criterion for tip vortex-core bursting is how much of the tip vortex-line segment being considered has passed beneath the blade (see Subsection 3.2.4). The standard value of DVB = 0.0 implies that all of the segment has passed beneath the blade, DVB = - 0.5 implies that only half of the segment has done so, and DVB = + 0.5 implies that the segment is one-half segment length behind the blade. The results at 85% radius for DVB = + 0.5 and DVB = - 0.5 are compared in Fig. 46. For the results with the standard value of DVB = 0.0, see the solid line in Fig. 40. The DVB = 0.0 and the DVB = + 0.5 results are similar, while the DVB = - 0.5 results are quite different. It is hard to justify DVB = - 0.5 physically, indeed DVB = + 0.5 is probably the most realistic value. The current version of LDS-73 requires that any particular segment be 100% burst or 100% not burst. It would be desirable to make provision for partially bursting a segment and more research should be done on just when a segment bursts.

5.3.4 Maximum Incremental Angle of Attack

There is a provision for a cutoff on the incremental angle of attack, due to any particular vortex-line segment (Subsection 3.2.7). This limit $\Delta\alpha_{max}$ is 0.05 radians for the standard case. It was thought that $\Delta\alpha_{max}$ might have an effect similar to tip vortex-core bursting and thus might be an alternative explanation for the small measured airloads peaks. Distorted wake airloads were computed for $\Delta\alpha_{max}$ of 0.03, 0.05, and 9.0 radians both with and without bursting. No effect was observed. Further investigation revealed that the maximum $\Delta\alpha$ never reached 0.03 radians for these cases and that $\Delta\alpha_{max}$ would need to be less than 0.02 radians to have any effect. In the no-lifting surface solution case, $\Delta\alpha$ does become as large as 0.075 radians; however, $\Delta\alpha_{max}$ is not intended to replace the lifting-surface solution. Thus, $\Delta\alpha_{max}$ is not important unless values smaller than 0.02 radians can be justified.

5.3.5 Tip Vortex Roll-Up

The vortex sheet plus vortex-line tip vortex model is used to represent tip vortex roll-up (Subsection 2.5.3). The alternative is instantaneous roll-up (i.e., a vortex-line model for the tip vortex). The effect of instantaneous roll-up on a rigid wake airloads computation for the four-blade, $\mu = 0.18$ case at 95% radius is shown in Fig. 47a. The effect of instantaneous roll-up on a distorted wake airloads computation for the same case, but with no bursting, is shown in Fig. 47b, at 85% radius. The effect on the rigid-wake case, with bursting included, is small; however, there is a substantial effect on the distorted wake case, without bursting. Thus, it is concluded that tip vortex roll-up can have a significant effect on the airloads in some cases. However, tip vortex roll-up cannot replace the propagation of tip vortex-core bursting since roll-up is included in the results shown in Fig. 43.

5.3.6 Tangentially-Induced Velocity

The component of the induced velocity which is tangential to the rotor blade chord is normally neglected compared to the sum of the rotational velocity ($\eta\Omega R$) and the free-stream velocity ($\mu\Omega R \sin\psi$). To test the validity of this assumption both rigid wake and distorted wake airloads were computed including the tangentially-induced velocity. In the rigid-wake case, the effect was negligible. In the distorted-wake case, the effect was small, except at the 85% radius (Fig. 48). The tangentially-induced velocities which produced the effect shown are only two or three percent of the total tangential velocity. Their effect is magnified by first harmonic flapping, however, because the tangentially-induced velocity consists primarily of a $\sin\psi$ term, in this case. Thus tangentially-induced velocities can be significant in special cases.

5.3.7 Effect of Distortion on Airloads

A direct comparison of rigid vs. distorted wake airloads for the four-blade $\mu = 0.18$ case is given in Fig. 2, at 95% radius. To compare the results at other radial stations match the solid line in Fig. 39 (rigid wake) with the solid line in Fig. 40 (distorted wake) for the 85% radius and match the solid line in Fig. 49 (rigid wake) with the solid line in Fig. 50 (distorted wake) for the 75% radius. The overall agreement with the measured data is about the same for this case.

If the effects of tip vortex-core bursting are eliminated and instantaneous roll-up is assumed, the distorted wake case (Fig. 50) becomes much worse than the rigid-wake case (Fig. 49), at the 75% radius. The key factor in this is the propagation of tip vortex-core bursting, as shown in Subsection 5.3.3. This effect is much less significant at the outer radial stations. If the lifting-surface solution is eliminated, the distorted-wake case (Fig. 40) again becomes much worse than the rigid-wake case (Fig. 39). This effect occurs at all radial stations (Subsection 5.3.1).

To check these results for another case, a six-blade, $\mu = 0.108$ case was run. The rigid vs. distorted tip vortex-geometry comparison for this case is shown in Fig. 37a ($\psi_s = 180°$) and in Fig. 37b ($\psi_s = 0°$). The corresponding airloads comparison is shown in Fig. 51, at the 85% radius. These airloads were computed using the same standard inputs as the four-blade case (Table 4), with an adjustment of the initial vortex-core size due to the different aspect ratio of the rotor blades. The distorted wake result is substantially better than the rigid-wake result at the 85% radius shown. However, the comparison at other radial stations is less favorable, making the overall comparison only somewhat in favor of the distorted wake.

5.4 Conclusions

The fundamental conclusion is that, over the forward portion of the rotor, the actual location of the tip vortex is much closer to the rotor than the rigid-wake assumption would predict. However, a lack of understanding of close blade-tip vortex interactions causes the computed airloads to be substantially larger than measured flight test data. This error tends to compensate for the rigid-wake assumption's error in the location of the tip vortex over the forward portion of the rotor, with the result that rigid-wake airloads computations agree fairly well with measured airloads. This same error causes airloads computed using the more accurate distorted tip vortex geometry to be too large.

A study was made of various possible close blade-tip vortex interaction effects. The largest one is the influence of the extra wake trailed during a close blade-vortex interaction on the airloads on that blade. This effect is well represented by Johnson's lifting-surface solution (Subsections 2.5.2 and 3.2.4). However, this effect is not sufficient to make the distorted wake airloads agree with the measured airloads. Thus, other effects were investigated.

The use of a sufficiently large tip vortex-core radius was found to reduce the distorted wake airloads by the necessary amount. However, the vortex-core radius required to achieve this result is about twenty times larger than that observed experimentally. This led to the concept of tip vortex-core bursting due to a close blade-tip vortex interaction. However, bursting alone does not explain the excessive distorted wake airloads at the more inboard radial stations such as the 75% radius. This led to the concept of propagation of bursting. Once a close blade-tip vortex interaction initiates bursting it is assumed to propagate along the tip vortex, ahead of the blade. This propagation of bursting is generally successful in reducing the distorted wake airloads to levels approximating those of the measured airloads.

Since tip vortex-core bursting is still controversial and propagation of bursting is speculative, a search was made for alternative explanations. Ham [54,55] has suggested that local separation limits the maximum incremental angle of attack due to any close blade-tip vortex interaction. The values of this limiting angle-of-attack increment required to have any effect on the case test are smaller than can be justified, however. The effects of tip vortex roll-up and of tangentially-induced velocities were also investigated. They both have modest effects on the airloads but they are not large enough or of the proper character to replace tip vortex-core bursting.

To summarize, the only satisfactory explanation found for the discrepancies between distorted wake airlaods and measured airloads is a combination of a lifting-surface solution and tip vortex-core bursting with propagation. The lifting-surface solution is satisfactory; however, the bursting model is highly simplified because there is as yet no basis for a better model. The distorted wake airloads computed using these techniques are not much more accurate than rigid wake airloads. This is probably due to an inadequate understanding of bursting and its propagation. However, it is possible that some other close blade-tip vortex interaction effect, no considered here, may also be important.

To make both airloads and wake geometry computations less expensive and hence more widely useful, a simplified wake model was developed which gives satisfactory results without great expense (Section 2). Special updating techniques were developed for computing the tip vortex distortion which reduce the expense by more than an order of magnitude (Subsection 4.2).

5.5 Recommendations

The principal recommendation is that a complete theoretical and experimental study of the close blade-tip vortex interaction problem be made, leading to a better understanding of the physics of the problem and to suitable mathematical models. Special emphasis should be given to the effect of close blade-tip vortex interactions on the tip vortex, including the possibility of tip vortex-core bursting. The possibility that bursting might propagate ahead of the blade which initiated it should be considered and the properties of a burst vortex need to be investigated so that a better model can be developed. The word bursting is used rather loosely here. It is generalized to include any change in the nature of the tip vortex, due to a close blade-vortex interaction, which changes the airloads induced by that tip vortex (see Subsection 2.5.1).

The formation and roll-up of the tip vortex should be studied as well. This is moderately important for airloads computations and it is thought to be the key to successful computation of the hovering tip vortex geometry. A study of the structure of the inboard trailing wake might yield useful results, but this is clearly less important than the previous recommendations.

Finally, various recommendations for improving the details of the wake model, the airloads computation, and the wake geometry computation are made in the main body of this report.

References

1. Meijer-Drees, J. "A Theory of Air Flow Through Rotors and its Application to Some Helicopter Problems". Journal Helicopter Assoc. of Great Britain, Vol. 3, No. 2, 1949.

2. Mangler, R.W. and Squire, H.B. "The Induced Velocity Field of a Rotor". Great Britain R & M 2642, 1953.

3. Castles, Walter, Jr. and DeLeeuw, Jacob Henri. "The Normal Component of the Induced Velocity in the Vicinity of a Lifting Rotor and Some Examples of its Application". U.S. NACA TR 1184, 1954.

4. Heyson, Harry H. and Katzoff, S. "Induced Velocities Near a Lifting Rotor with Nonuniform Disk Loading". U.S. NACA TR 1319, 1957.

5. Miller, R.H. "Rotor Blade Harmonic Air Loading". IAS Paper 62-82, Jan. 1962.

6. Miller, R.H. "On the Computation of Airloads Acting on Rotor Blades in Forward Flight". American Helicopter Society Journal, Vol. 7, No. 2, April 1962.

7. Miller, R.H. "Rotor Blade Harmonic Airloads". AIAA Journal, Vol. 2, No. 7, July 1964.

8. Miller, R.H. "Theoretical Determination of Rotor Blade Harmonic Airloads". Massachusetts Institute of Technology, Aeroelastic and Structures Research Laboratory, ASRL TR 107-2, August 1964.

9. Miller, R.H. "Unsteady Air Loads on Helicopter Rotor Blades". Royal Aeronautical Society Journal, Vol. 68, No. 640, April 1964.

10. Scully, M.P. "Approximate Solutions for Computing Helicopter Harmonic Airloads". Massachusetts Institute of Technology, Aeroelastic and Structures Research Laboratory, ASRL TR 123-2, December 1965.

11. Piziali, R.A. and DuWaldt, F.A. "A Method for Computing Rotory Wing Airload Distributions in Forward Flight". U.S. Army TRECOM TCREC TR 62-44, November 1962.

12. Piziali, R.A. and DuWaldt, F. "Computation of Rotory Wing Harmonic Airloads and Comparison with Experimental Results". American Helicopter 18th Annual Forum (1962).

13. Davenport, F.J. "A Method for Computation of the Induced Velocity Field of a Rotor in Forward Flight, Suitable for Application to Tandem Rotor Configurations". American Helicopter Society Journal, Vol. 9, No. 3, July 1964.

14. Piziali, R.A. "A Method for Predicting the Aerodynamic Loads and Dynamic Response of Rotor Blades". USAAVLABS, TR 65-74, January 1966.

15. Daughaday, H. and Piziali, R.A. "An Improved Computation Model for Predicting the Unsteady Aerodynamic Loads of Rotor Blades". American Helicopter Society Journal, Vol. 11, No. 4, October 1966.

16. Madden, Paul A. "Angle-of-Attack Distribution of a High Speed Helicopter". American Helicopter Society Journal, Vol. 12, No. 2, April 1967.

17. Ham, Norman D. "An Experimental Investigation of the Effect of a Non-rigid Wake on Rotor Blade Airlaods in Transition Flight". Cornell Aeronautical Laboratory/ TRECOM Symposium, 1963.

18. Scully, M.P. "Helicopter Vortex Wake Geometry". Massachusetts Institute of Technology, S.B. Thesis, June 1964.

19. Scully, M.P. "A Method of Computing Helicopter Vortex Wake Distortion". Massachusetts Institute of Technology, Aeroelastic and Structures Research Laboratory, ASRL TR 138-1, June 1967.

20. Crimi, P. "Theoretical Prediction of the Flow in the Wake of a Helicopter Rotor". Cornell Aeronautical Laboratory, Report No. BB-1944-S-1, September 1965.

21. Sadler, S.G. "A Method for Predicting Helicopter Wake Geometry, Wake-Induced Flow and Wake Effects on Blade Airloads". 1971 American Helicopter Society 27th Annual Forum Preprint 523.

22. Landgrebe, Anton J. "An Analytical Method for Predicting Rotor Wake Geometry". American Helicopter Society Journal, Vol. 14, No. 4, October 1969, pp. 20-32.

23. Landgrebe, Anton J. "An Analtyical and Experimental Investigation of Helicopter Rotor Hover Performance and Wake Geometry Characteristics". USAAMRDL TR 71-24, June 1971.

24. Widnall, Sheila E. "The Stability of a Helical Vortex Filament". Journal of Fluid Mechanics, Vol. 54, Pt.4, 22 Aug. 1972

25. Goldstein, S. "On the Vortex Theory of Screw Propellors". Proceedings - Royal Society of London, Vol. A 123, No. A 792, 1929.

26. Lock, C.N.H. "The Application of Goldstein's Theory to the Practical Design of Airscrews". Great Britain R & M 1377, November 1930.

27. Murray, James C. "Generalized Actuator Disk Theory". Massachusetts Institute of Technology, S.M. Thesis, June 1968.

28. Erickson, J.C. Jr. "Theoretical and Experimental Investigations of V/STOL Propellor Operation in a Static Condition". USAAVLABS TR 69-55, October 1969.

29. Clark, D.R. and Landgrebe, Anton J. "Wake and Boundary Layer Effects in Helicopter Rotor Aerodynamics". AIAA Paper 71-581,

30. Clark, David R. and Leiper, Albert C. "The Free Wake Analysis - A Method for the Prediction of Helicopter Rotor Hovering Performance". American Helicopter Society Journal, Vol. 15, No. 1, January 1970.

31. Landgrebe, Anton, J. "The Wake Geometry of a Hovering Rotor and its Influence on Rotor Performance". American Helicopter Society Preprint 620 (28th Annual Forum May 1972).

32. Rorke, James B. and Wells, Clifford D. "The Prescribed Wake Momentum Analysis". Cornell Aeronautical Laboratory/ AVLABS Symposium, Vol. 1, 1969.

33. Landgrebe, Anton J. and Cheney, Marvin C. "Rotor Wakes - Key to Performance Prediction". Symposium on Status of Testing and Modeling Techniques for V/STOL Aircraft, AHS Mid-East Region, October 1972.

34. Glauert H. "The Elements of Aerofoil and Airscrew Theory". Cambridge at the University Press,(1959).

35. Lamb, Horace. Hydrodynamics. Dover Publications, New York, N.Y.,1932.

36. Surendraiah, M. "An Experimental Study of Rotor Blade-Vortex Interaction". NASA CR-1573, May 1970.

37. Corsiglia, V.R., Schwind, R.G., and Chigier, N.A. "Rapid Scanning, 3D Hot-Wire Anemometer Surveys of Wing-Tip Vortices". Journal of Aircraft, Vol. 10, No. 12, Dec. 1973. p. 752

38. Rorke, J.B. and Moffitt, R.C. "Wind Tunnel Simulation of Full-Scale Vortices". NASA CR-2180, March 1973.

39. Cook, C.V. "The Structure of the Tip Vortex". NATO AGARD-CP-111, paper No. 3, September 1972.

40. Bliss, Donald B. "The Dynamics of Curved Rotational Vortex Lines". Massachusetts Institute of Technology S.M. Thesis, September 1970.

41. Betz, A. "Behavior of Vortex Systems". NACA TM 713, June 1933.

42. Chigier, N.A. and Corsiglia, V.R. "Tip Vortices - Velocity Distributions". American Helicopter Society 27th Annual Forum Preprint 522, May 1971.

43. Orloff, K.L. and Grant, G.R. "The Application of a Scanning Laser Doppler Velocimeter to Trailing Vortex Definition". AIAA Paper 73-680, July 1973.

44. Biggers, J.C. and Orloff, K.L. "Laser Velocimeter Measurements of the Helicopter Rotor-Induced Flow Field". American Helicopter Society 30th Annual National Forum, Preprint 800, May 1974.

45. Boatwright, D.W. "Measurement of Velocity Components in the Wake of a Full-Scale Helicopter Rotor in Hover". USAAMRDL TR 72-33, August 1972.

46. Donaldson, C. duP., Snedeker, R.S., and Sullivan, R.D. "Calculation of the Wakes of Three Transport Aircraft in Holding, Takeoff, and Landing Configurations, and Comparison with Experimental Measurements". AFOSR TR-73-1594, March 1973.

47. Rossow, V.J. "On the Inviscid Rolled-Up Structure of Lift-Generated Vortices". Journal of Aircraft, Vol.10,No. 11, Nov.1973.

48. Bilanin, A. and Donaldson, C. duP. "Estimation of Velocities and Roll-Up in Aircraft Vortex Wakes". ARAP, Princeton, N.J. (to be published).

49. Scheiman, J. and Ludi, L.H. "Qualitative Evaluation of Effect of Helicopter Rotor-Blade Tip Vortex on Blade Airloads". NASA TN D-1637, 1963.

50. Scheiman, James. "A Tabulation of Helicopter Rotor-Blade Differential Pressures, Stresses, and Motions as Measured in Flight". NASA TM X-952, March 1964.

51. Beno, Edward A. "CH-53A Main Rotor and Stabilizer Vibrating Airloads and Forces". Sikorksy Engineering Report SER 65593, June 1970.

52. Johnson, Wayne R. "A Lifting Surface Solution for Vortex Induced Airloads and its Application to Rotary Wing Airloads Calculations". Massachusetts Institute of Technology, Aeroelastic and Structures Laboratory, ASRL TR 153-2, April 1970.

53. Johnson, Wayne R. "A Lifting-Surface Solution for Vortex-Induced Airloads". AIAA Journal, Vol. 9, No. 4, April 1971.

54. Ham, Norman D. "Some Preliminary Results from an Investigation of Blade-Vortex Interaction". American Helicopter Society Journal, Vol.19, No. 2, April 1974.

55. Ham, Norman D. "Summary of MIT Research on Dynamic Stall and Blade-Vortex Interaction 1971-1974". Massachusetts Institute of Technology, Aeroelastic and Structures Research Laboratory, ASRL TR 130-4, September 1974.

56. Bisplinghoff, R.L., Ashley, H., and Halfman, R.L. Aeroelasticity. Addison-Wesley, Reading, Mass. 1955.

57. Harris, F.D., Tarzanin, F.J., Jr., and Fisher, R.K. Jr. "Rotor High Speed Performance, Theory vs. Test". American Helicopter Society Journal, Vol. 15, No. 3, July 1970.

58. Gessow, A. and Myers, G.C. Jr. Aerodynamics of the Helicopter. Frederick Ungar Publishing Co., New York, N.Y. 1952.

59. Erickson, Gregg H. "Flow Visualization Study of Helicopter Tip Vortices Using a Four-Bladed Rotor System". Massachusetts Institute of Technology 16.62 Report, March 1970.

TABLE 1

NORMALIZATION RULES

TYPE OF VARIABLE	NORMALIZATION
Length	R = rotor radius (ft)
Time	Ω = rotor angular velocity (radians/sec)
Speed	$\Omega R = V_T$ = tip speed (ft/sec)
Angle	Radian measure
Circulation	$2\pi b\Omega R$ (ft^2/sec), where b = rotor blade semichord
Airload	$\rho c(\Omega R)^2/2$ (lb/in), where c = rotor blade chord

Note: Since the unit of length normally used here is feet, just using the normal values of ρ,Ωc, and ΩR will give (lb/ft) This must be divided by 12 to get (lb/in)

TABLE 2

STANDARD CASES

No. Blades	1	4	6
μ	0.10	0.18	0.108
λ	0.040	0.024	0.028
σ	0.035	0.0622	0.115
C_T	0.0027	0.0055	0.0060
μ tan i	0.0113	0.009	0.0009
θ_1 (radians)	0.0	- 0.14	- 0.14
Locke Number	1.085	9.7	12.6
ε	0.10	0.036	0.056
ΩR (ft/sec)	79.5	662	715
R(ft)	1.89	28	36
ρ (slug/ft^3)	0.00238	0.00216	0.00217
Reference	59	49	51

TABLE 3

EXECUTION TIMES*

BLADES	μ	m/M	DM	ℓ_{DM}	$n_{DM}(\psi)$	WG TIME	LDS TIME
1	.10	4/4	.002	12	6/3	39	12
4	.18	2/2	.002	12	6/3	32	30
4	.18	2/2	.001	12	6/3	37	30
4	.18	2/2	.0005	12	6/3	43	30
4	.18	2/2	.002	4	2	86	30
6	.108	4/4	.0028	12	6/3	177	70

---

*Times are CPU time in seconds on an IBM 370/168 running under release 21.7 of OS/MVT with release 21.6 FORTRAN G1. Most of the CALCOMP plot time is excluded from the WG times given. The WG times are for two iterations and the LDS times are for 5 iterations. However, the induced velocity influence coefficient matrix (QMAT) is only computed in the LDS computations.

TABLE 4

STANDARD AIRLOADS INPUTS

Vortex Core Radius (normalized by R)

- a) Initial value $\rho_k = 0.0025$
- b) Burst value $\rho_g = 0.100$
- c) Vortex sheet value $\rho_{VS} = 0.0025$

Vortex Core Bursting Parameters

- a) Distance from blade DVB = 0.0 segment lengths
- b) Distance below blade FVB = 20.0 vortex core radii

Bursting Propagation Parameters

- a) Propagation rate NPRP = 3 (increment bursting azimuth angle by one $\Delta\phi$ for every NPRP $\Delta\phi$ steps in age)
- b) Propagation range NRNG = 16 ,end propagation at $\phi = (\text{NRNG}-1)\Delta\phi$

Lifting-Surface Solution - included

Maximum incremental angle of attack $\alpha_m = 0.05$ radians

Vortex Sheet Plus Vortex-Line Tip-Vortex Model

Roll-up rate $f_t^D = 0.05$ (fraction of total tip vortex circulation which is in the vortex sheet decreases by f_t^D each $\Delta\phi$ step)

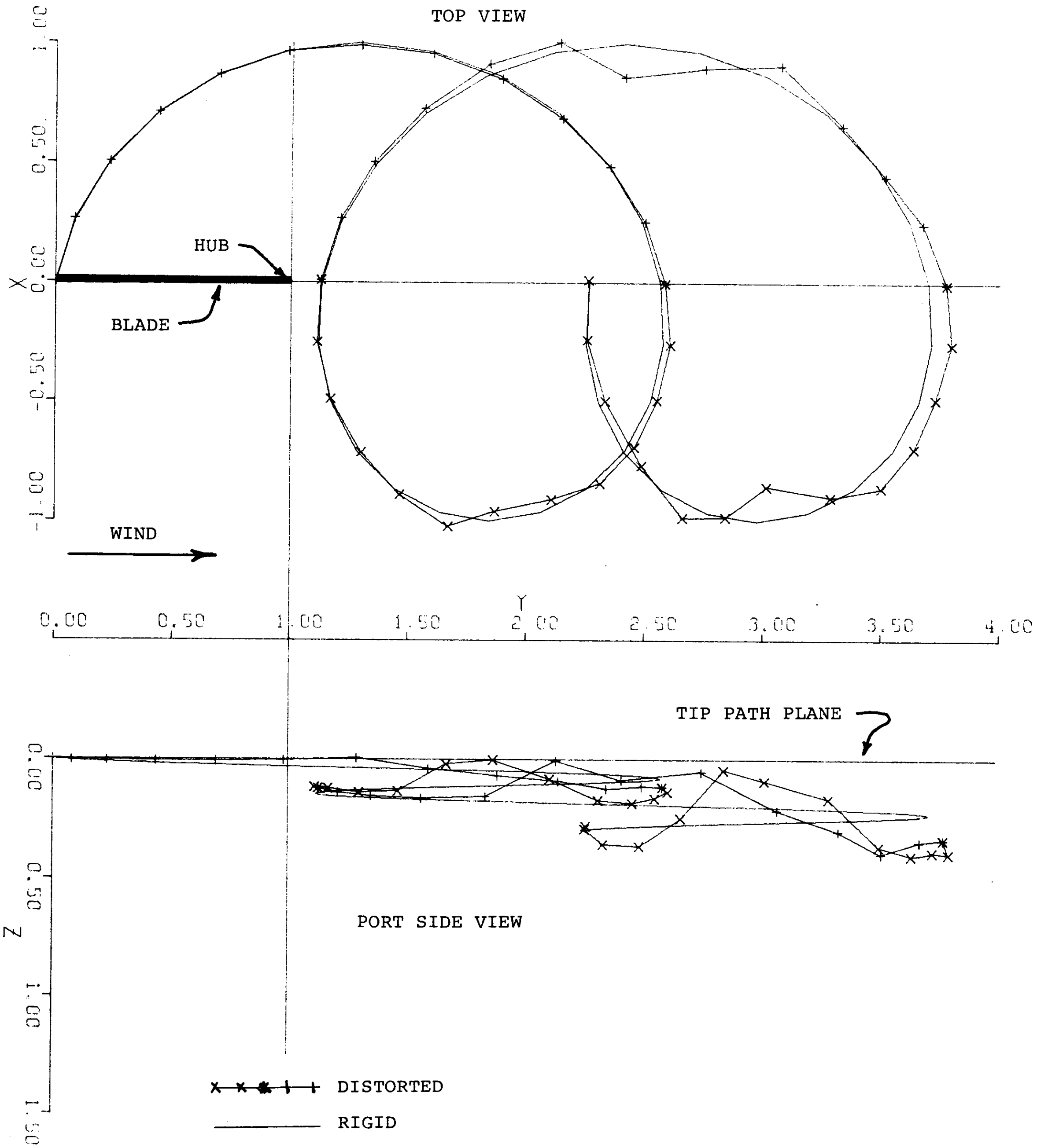

FIG. 1 RIGID VS. DISTORTED TIP VORTEX GEOMETRIES FOR ψ = 180° BLADE OF 4 BLADE ROTOR AT μ = 0.18

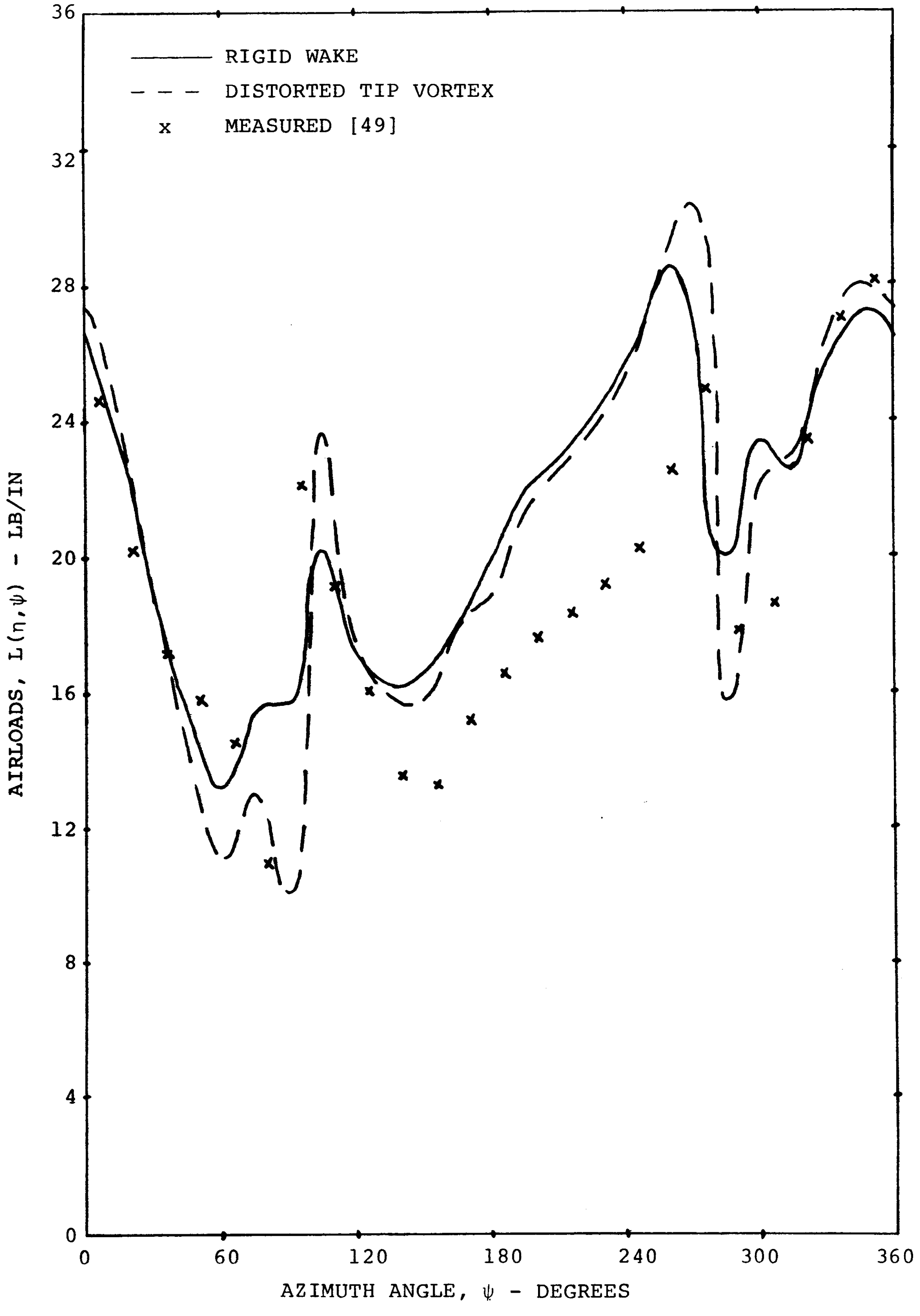

FIG. 2 RIGID VS. DISTORTED WAKE AIRLOADS, 95% RADIUS, 4 BLADE ROTOR, $\mu = 0.18$

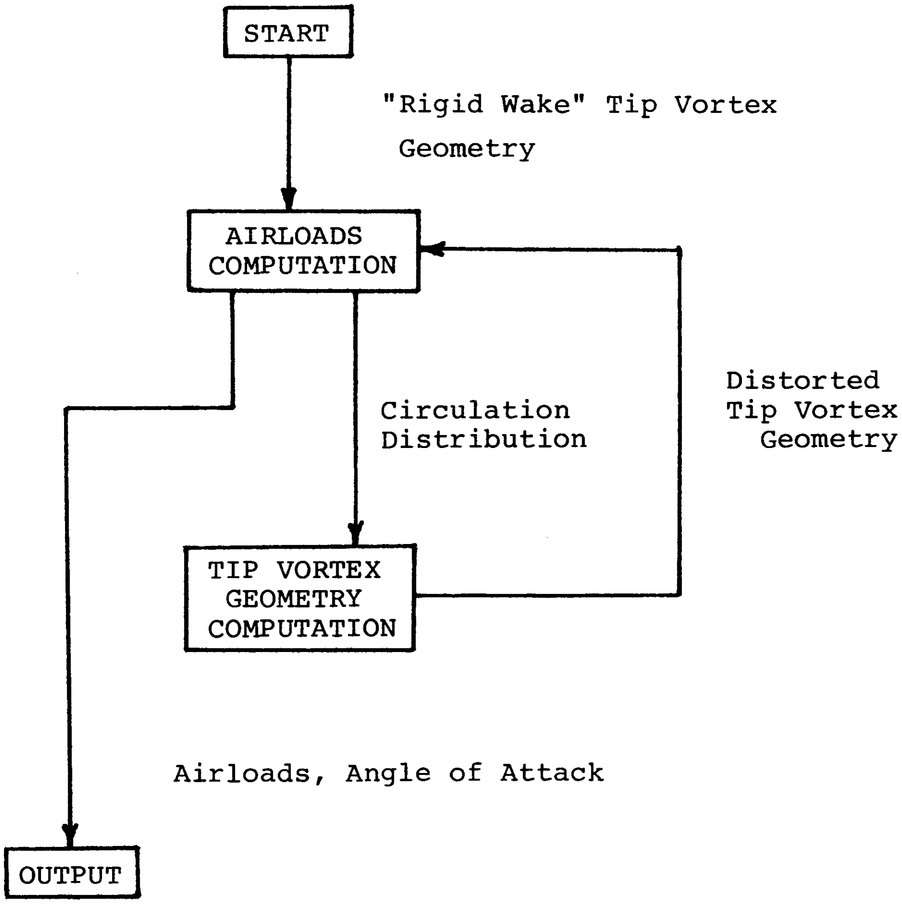

FIG. 3 BASIC TIP VORTEX GEOMETRY-CIRCULATION ITERATION

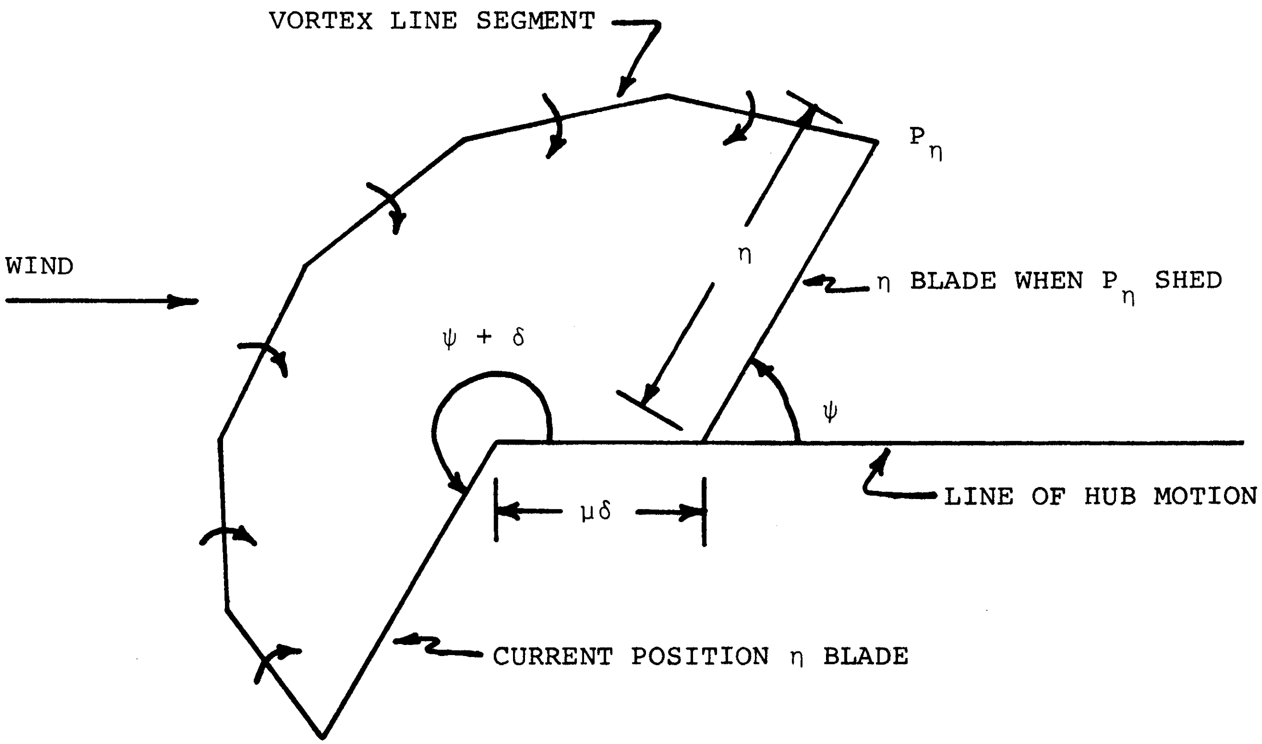

FIGURE 4 TOP (TPP) VIEW OF THE η BLADE AT TWO DIFFERENT TIMES: $\psi = 60°$, $\Delta\psi = 30°$, $\eta = 1.0$, $\delta = 180°$, $\zeta = 0°$, $\mu = .19$

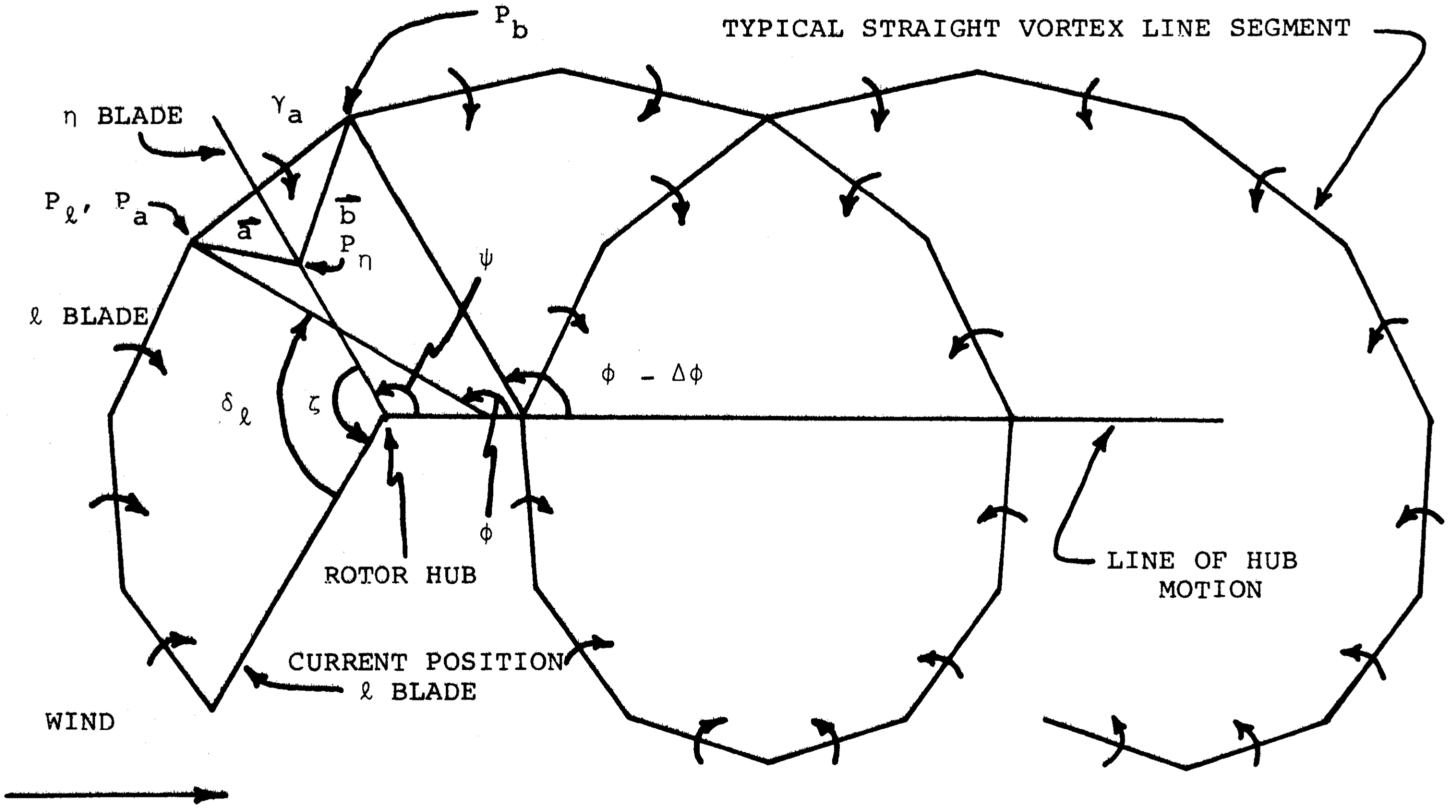

FIGURE 5: TOP (TPP) VIEW OF TWO RIGID WAKE SPIRALS OF THE TIP VORTEX OF ONE BLADE OF A THREE-BLADED ROTOR PLUS THE η BLADE FOR: $\psi = 120°$, $\eta = .50$, $\phi = 150°$, $\Delta\phi = 30°$, $\ell = 1.0$, $\delta_\ell = 90°$, $\zeta = 120°$, $n_b = 3$, $m = 2$, $\mu = .19$.

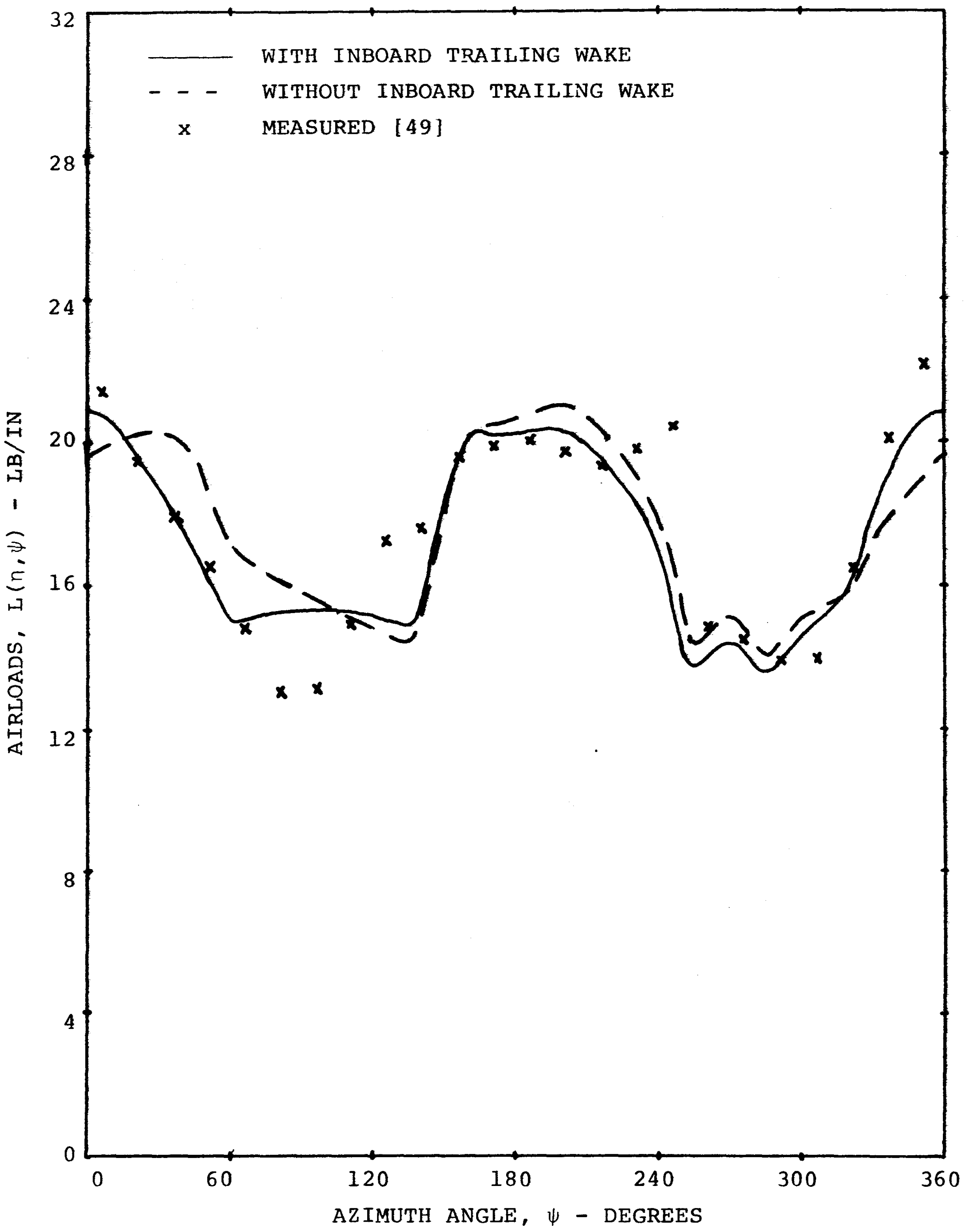

FIG. 6 RIGID WAKE AIRLOADS, WITH AND WITHOUT INBOARD TRAILING WAKE, 75% RADIUS, 4 BLADE ROTOR, $\mu = 0.18$

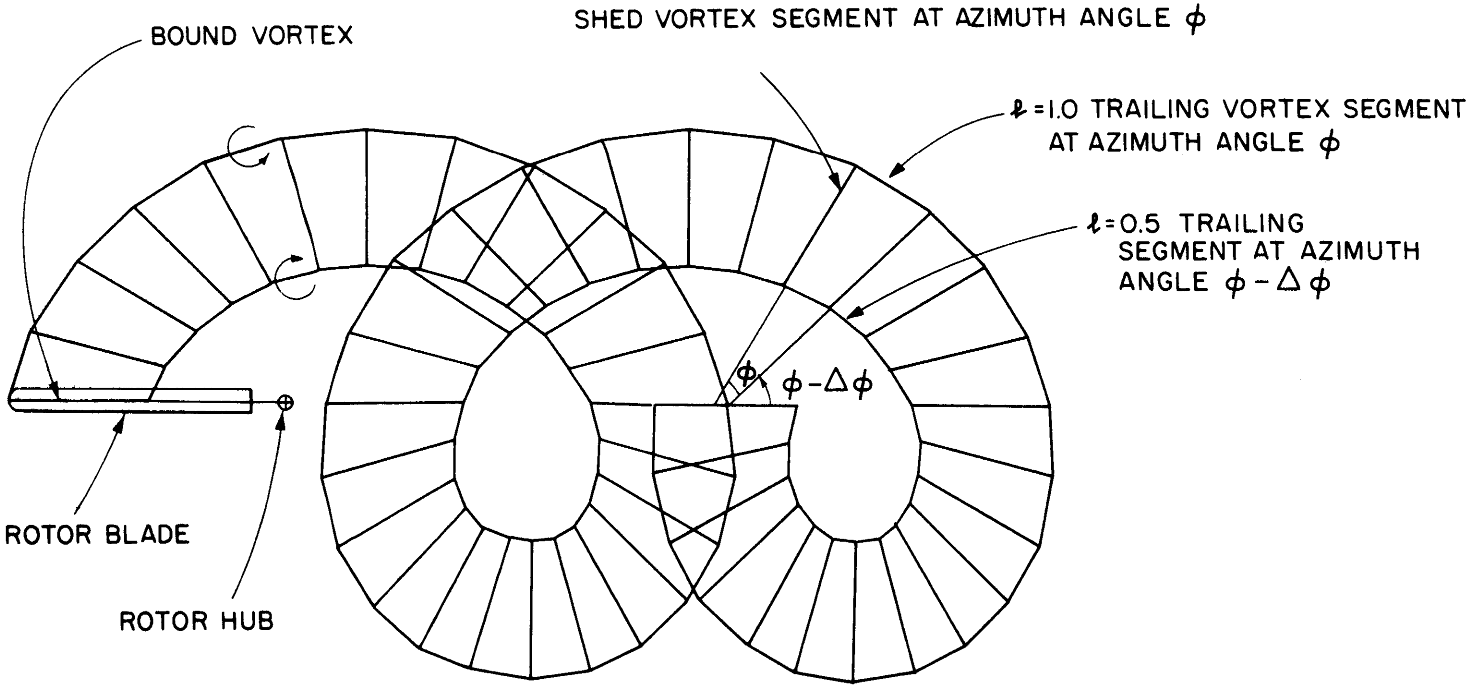

FIG. 7 BASIC WAKE MODEL OF WAKE OF ONE BLADE PROJECTED INTO TPP (RIGID WAKE)

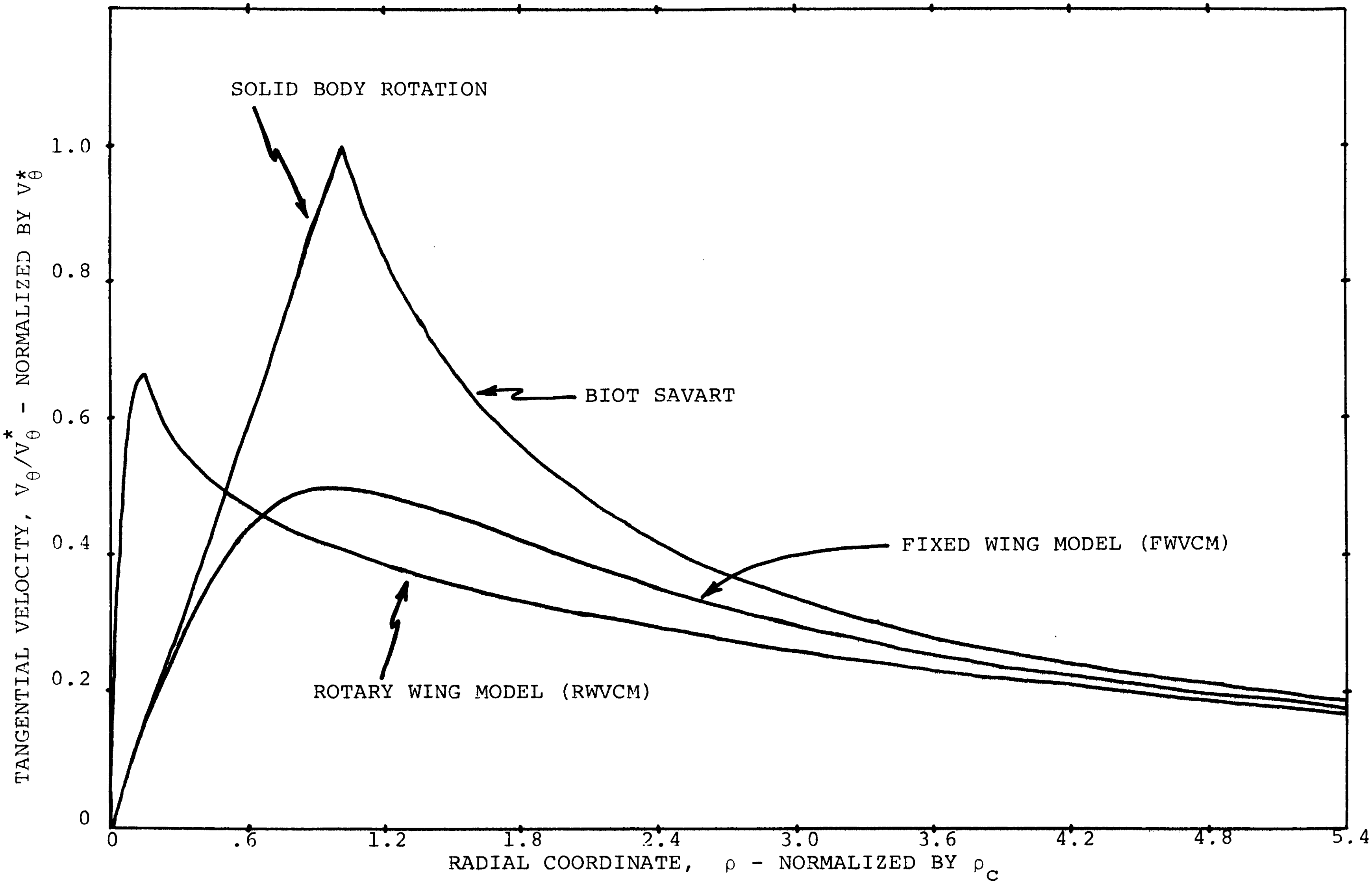

FIG. 8 COMPARISON OF VORTEX CORE MODELS

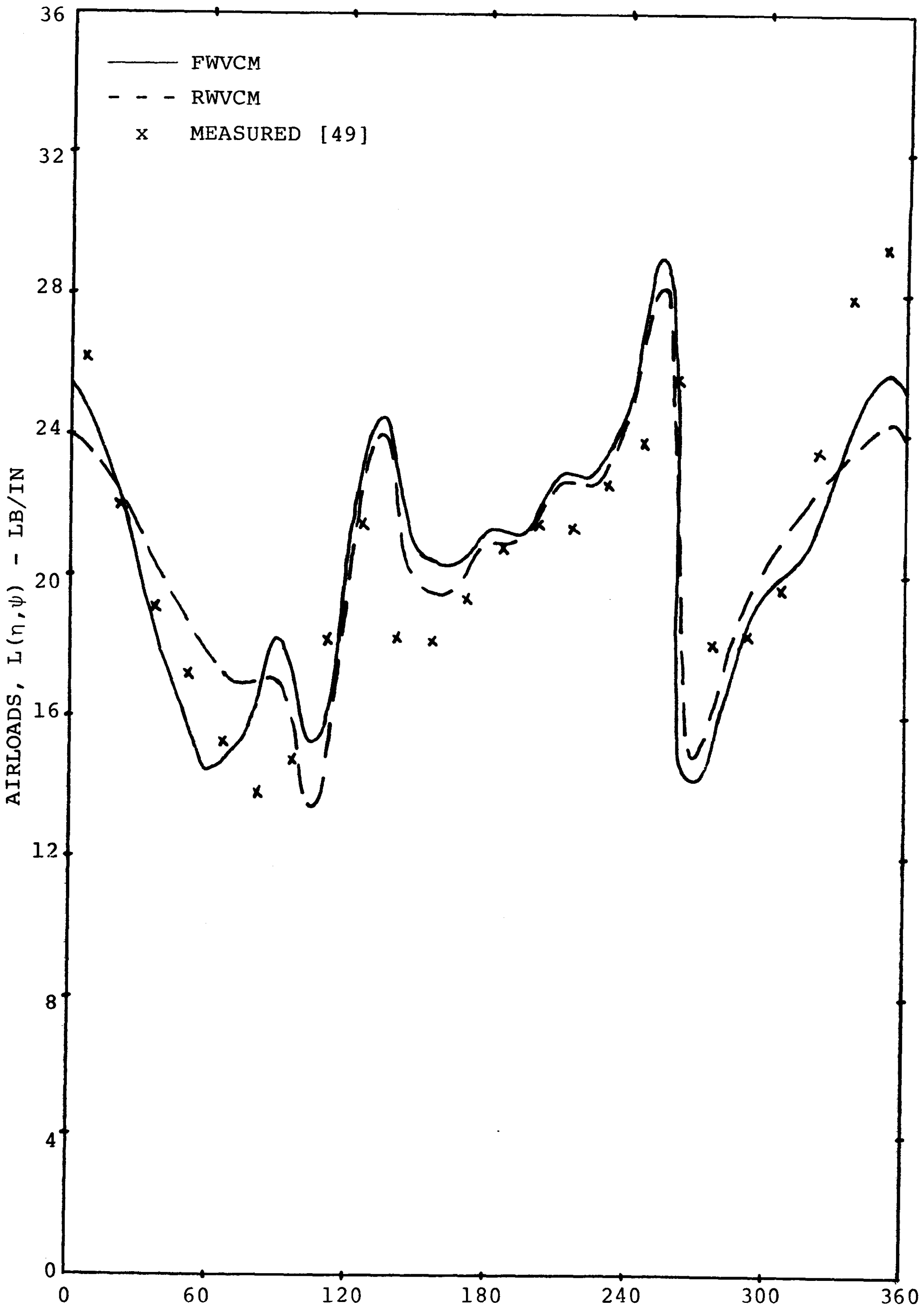

FIG. 9 DISTORTED WAKE AIRLOADS, COMPARISON OF VORTEX CORE MODELS, 85% RADIUS, 4 BLADE ROTOR, $\mu = 0.18$

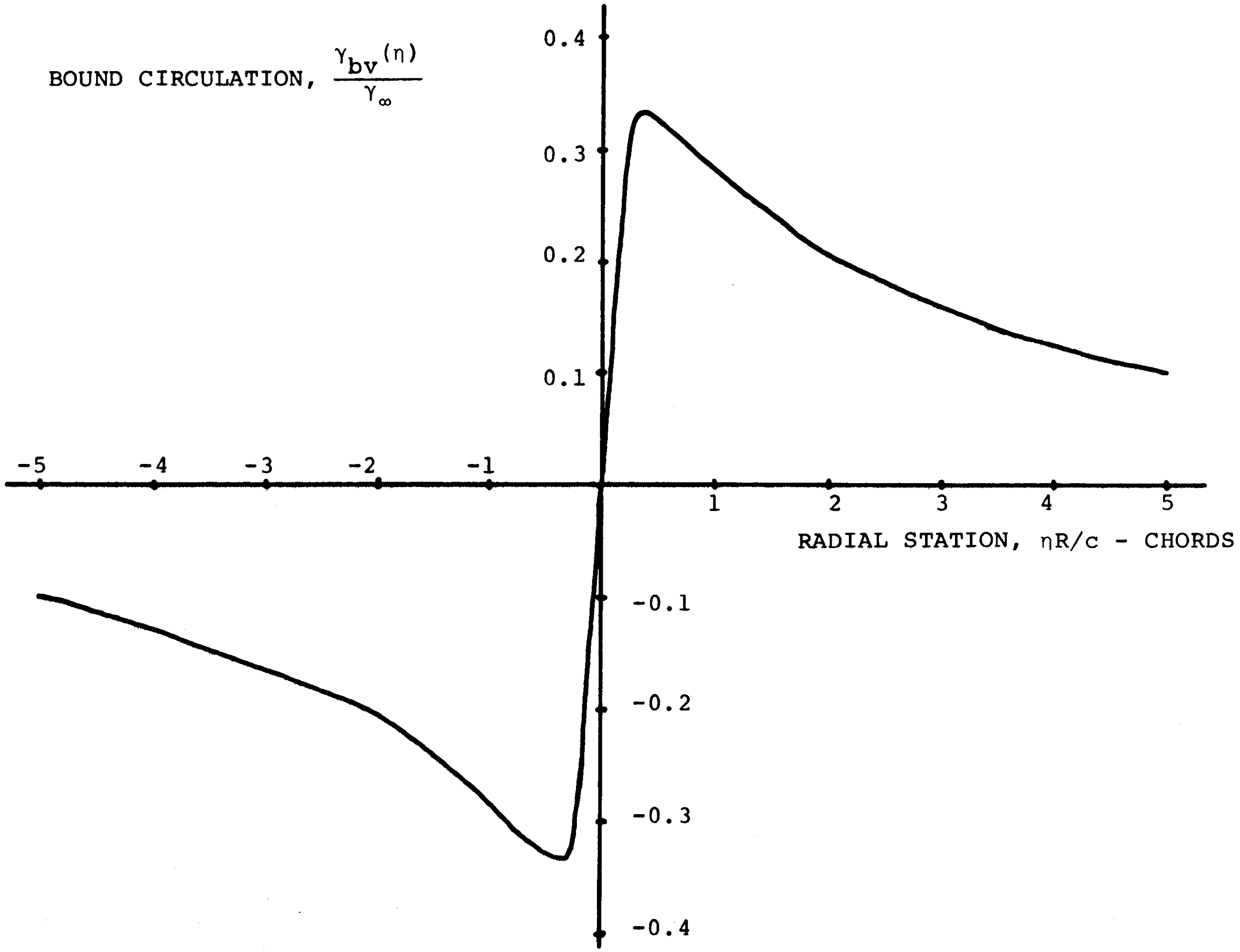

FIG. 10 TYPICAL SPANWISE DISTRIBUTION OF BOUND CIRCULATION $\gamma_{bv}(\eta)$ DUE TO PASSAGE OF A VORTEX LINE OF CIRCULATION γ_∞ A DISTANCE 0.1 CHORD BELOW THE BLADE [52]

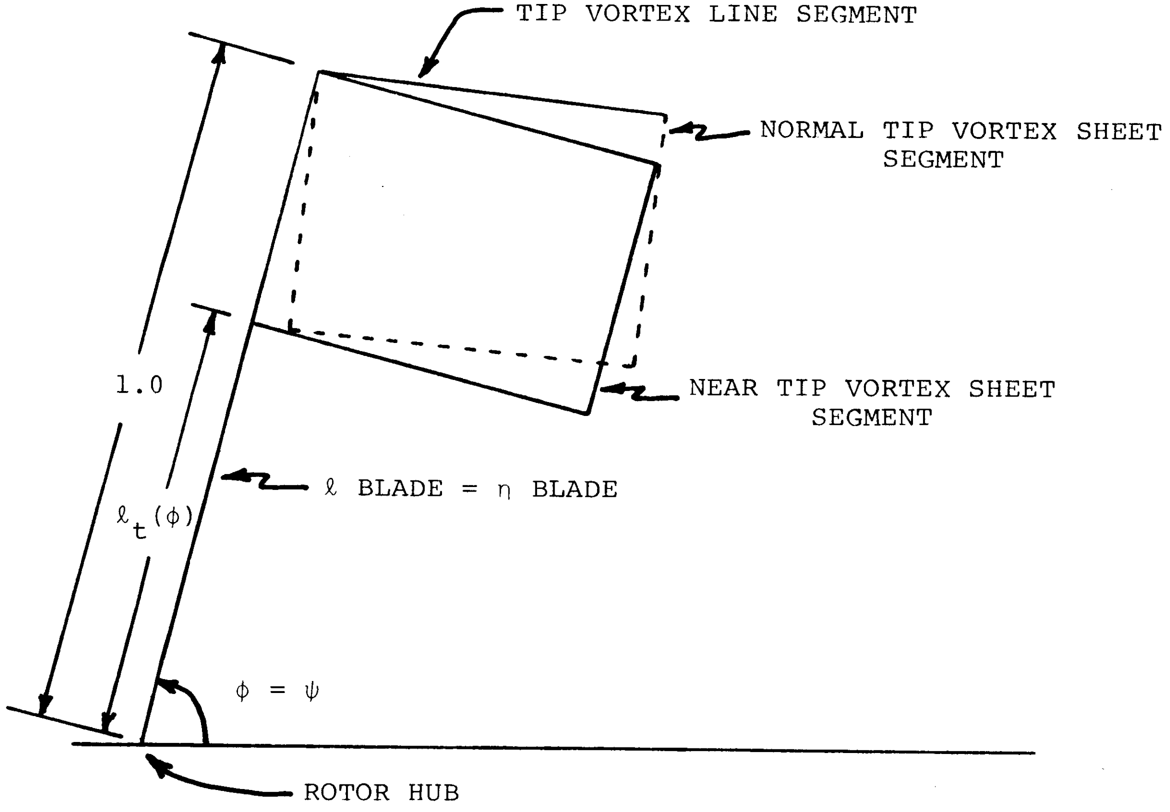

FIG. 11 TPP VIEW OF A NEAR TIP VORTEX SHEET SEGMENT COMPARED TO A NORMAL TIP VORTEX SHEET SEGMENT

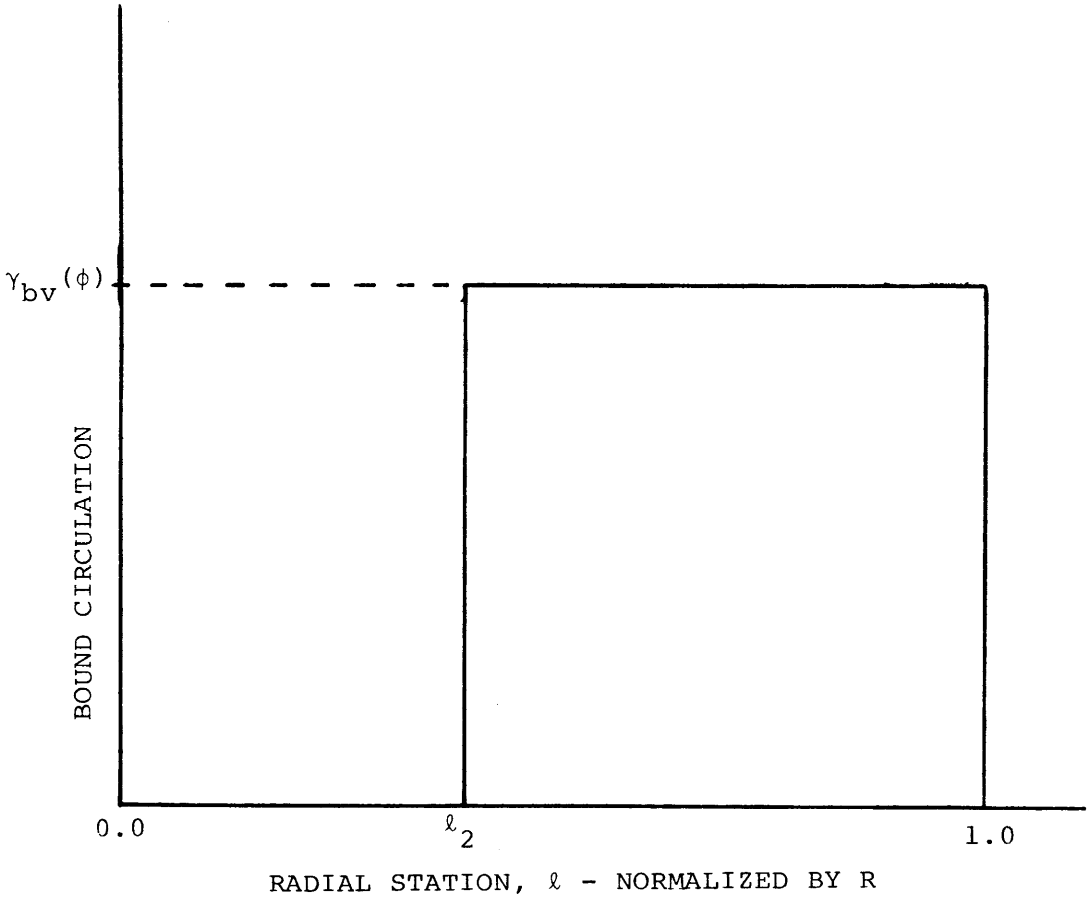

FIG. 12 BOUND CIRCULATION VS. RADIUS FOR SIMPLE BASIC WAKE MODEL

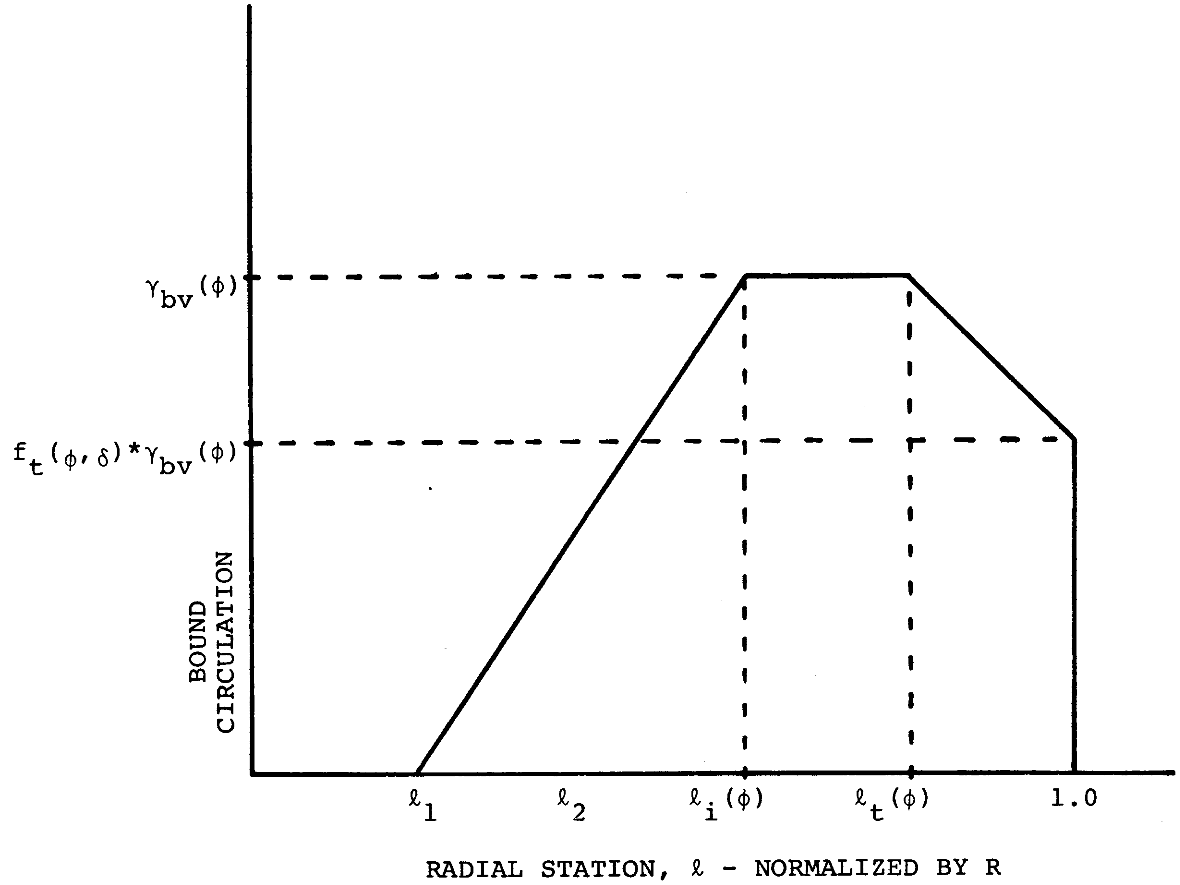

FIG. 13 BOUND CIRCULATION VS. RADIUS FOR BASIC WAKE MODEL WITH INBOARD TRAILING VORTEX SHEET AND TIP VORTEX SHEET PLUS LINE MODELS

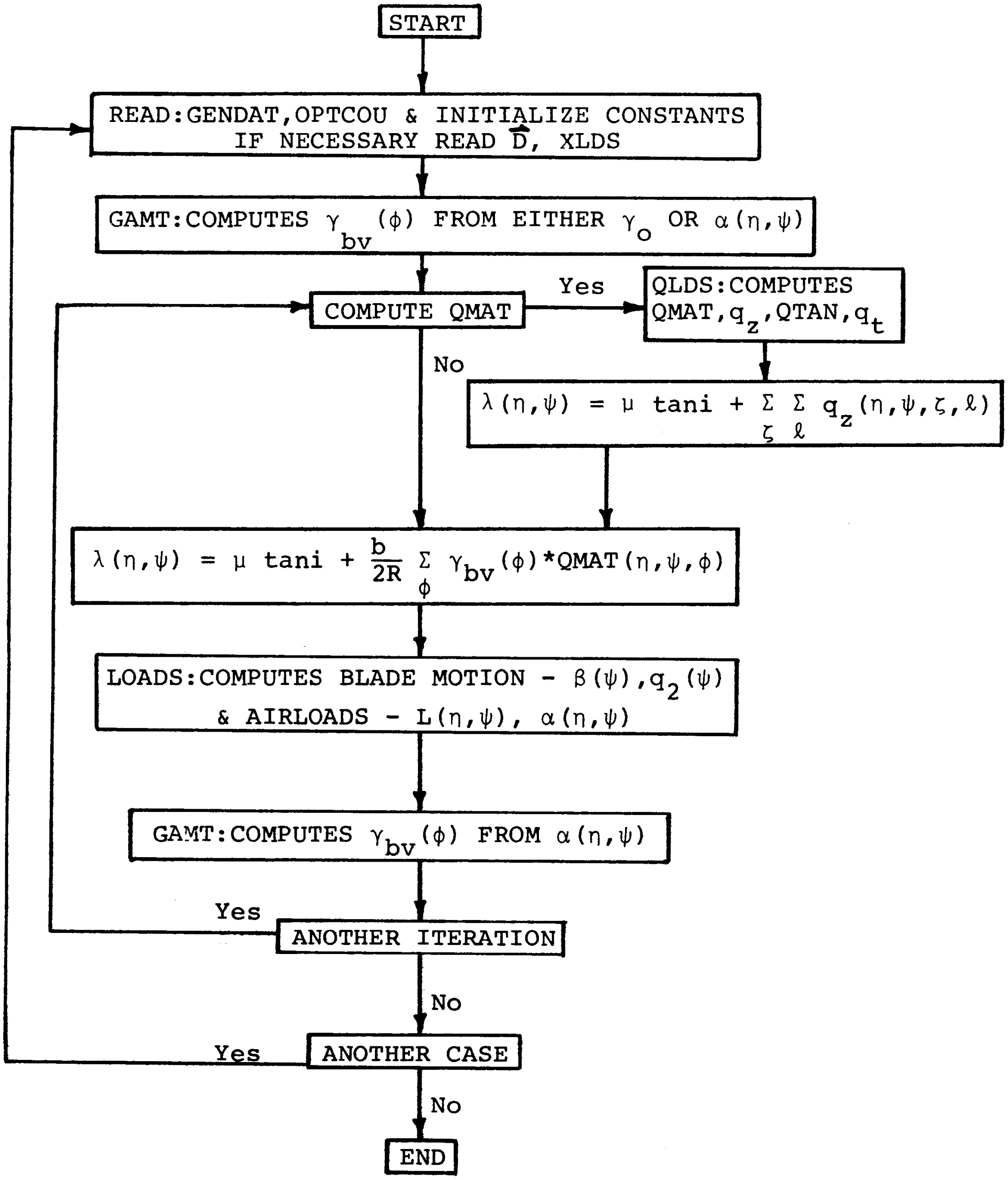

FIG. 14 SIMPLIFIED FLOW CHART FOR AIRLOADS PROGRAM LDS-73

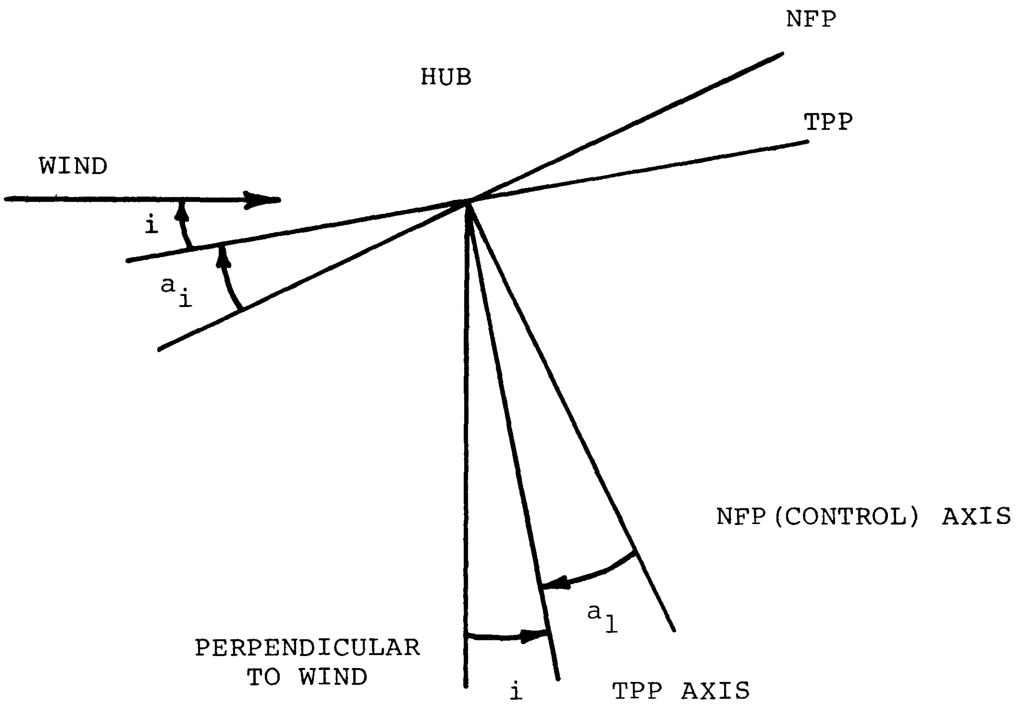

FIG. 15 RELATIONSHIP OF TIP PATH PLANE (TPP) AND NO-FEATHERING PLANE (NFP) TO RELATIVE WIND

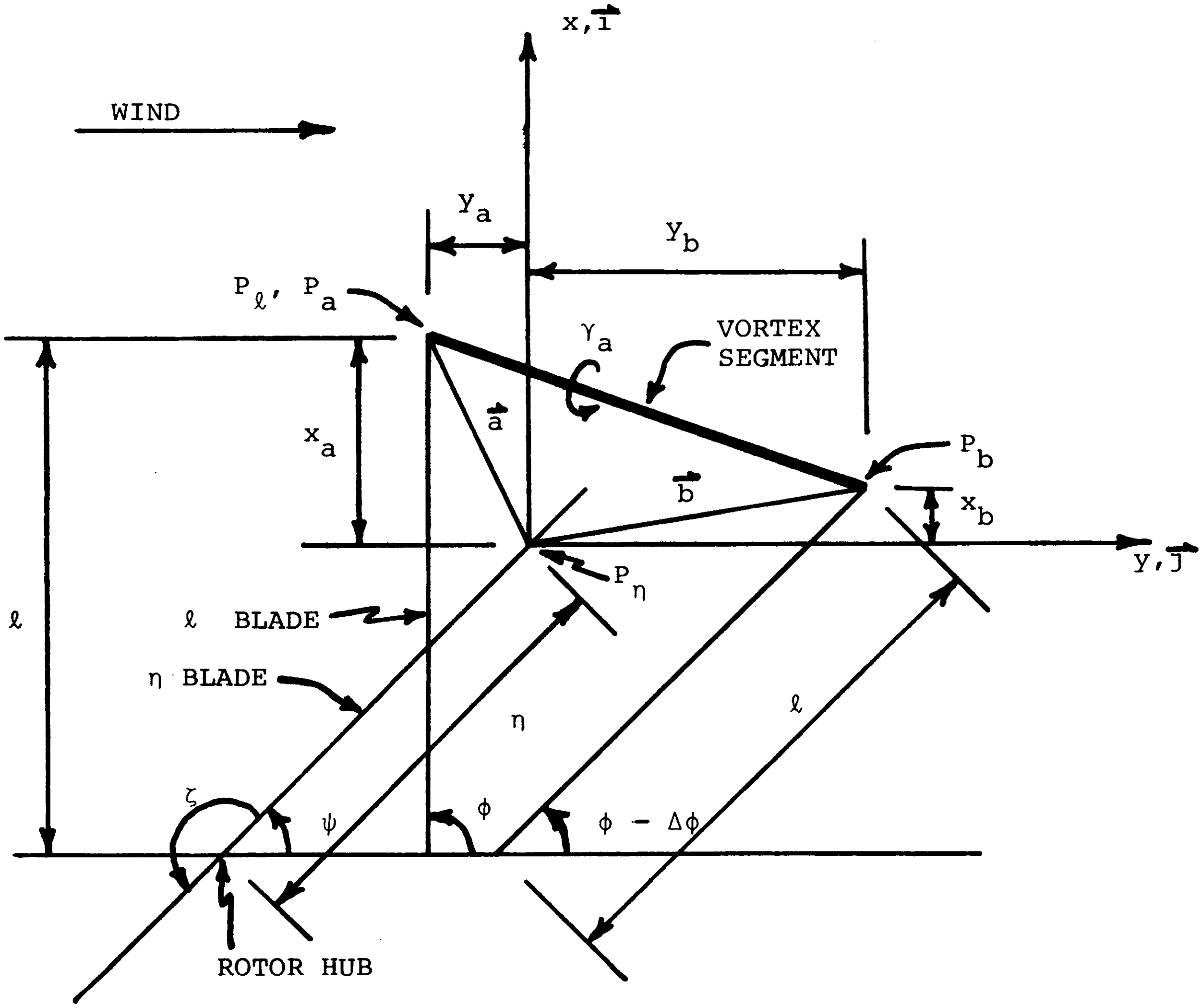

FIG. 16 TOP (TPP) VIEW OF A TYPICAL STRAIGHT VORTEX LINE SEGMENT $\psi = 45°$, $\eta = .835$, $\phi = 90°$, $\Delta\phi = 45°$, $\ell = 1.0$, $\delta = 135°$, $\zeta = 180°$, $n_b = 2$, $\mu = .255$

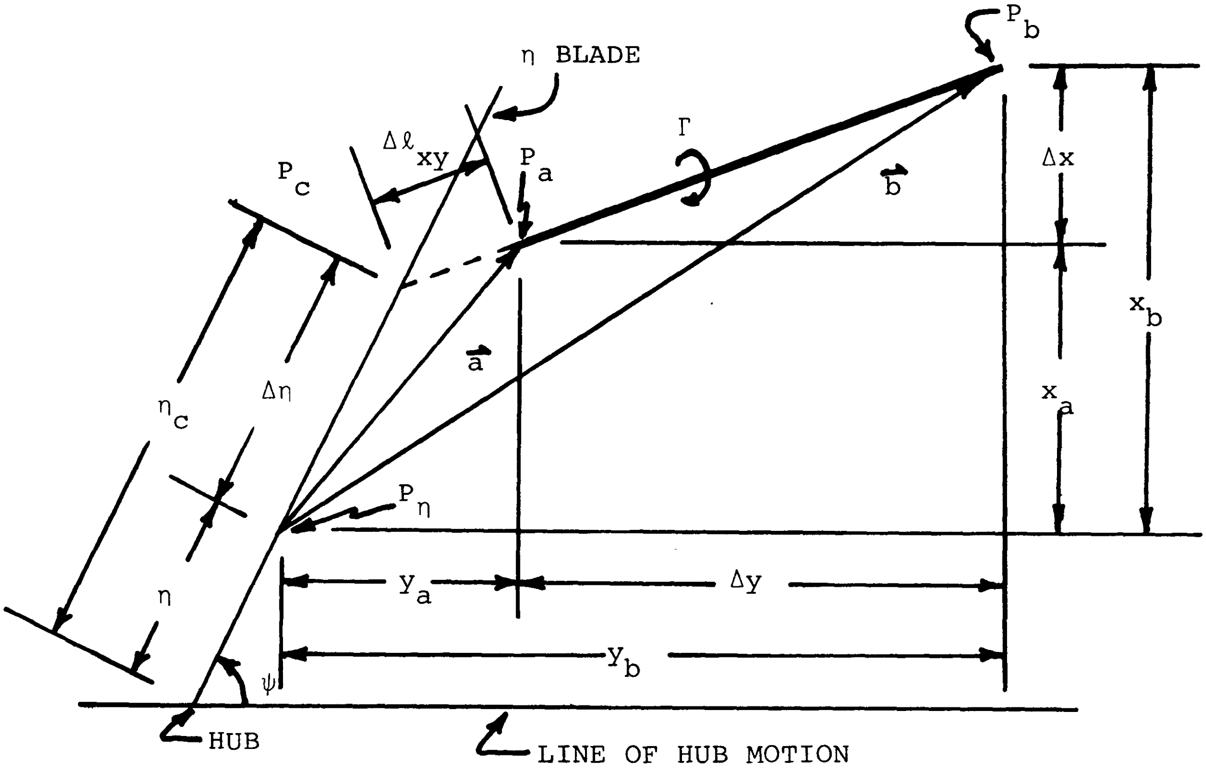

FIG. 17 TOP (TPP) VIEW OF BLADE-VORTEX LINE SEGMENT INTERACTION GEOMETRY, PROJECTED INTO TPP

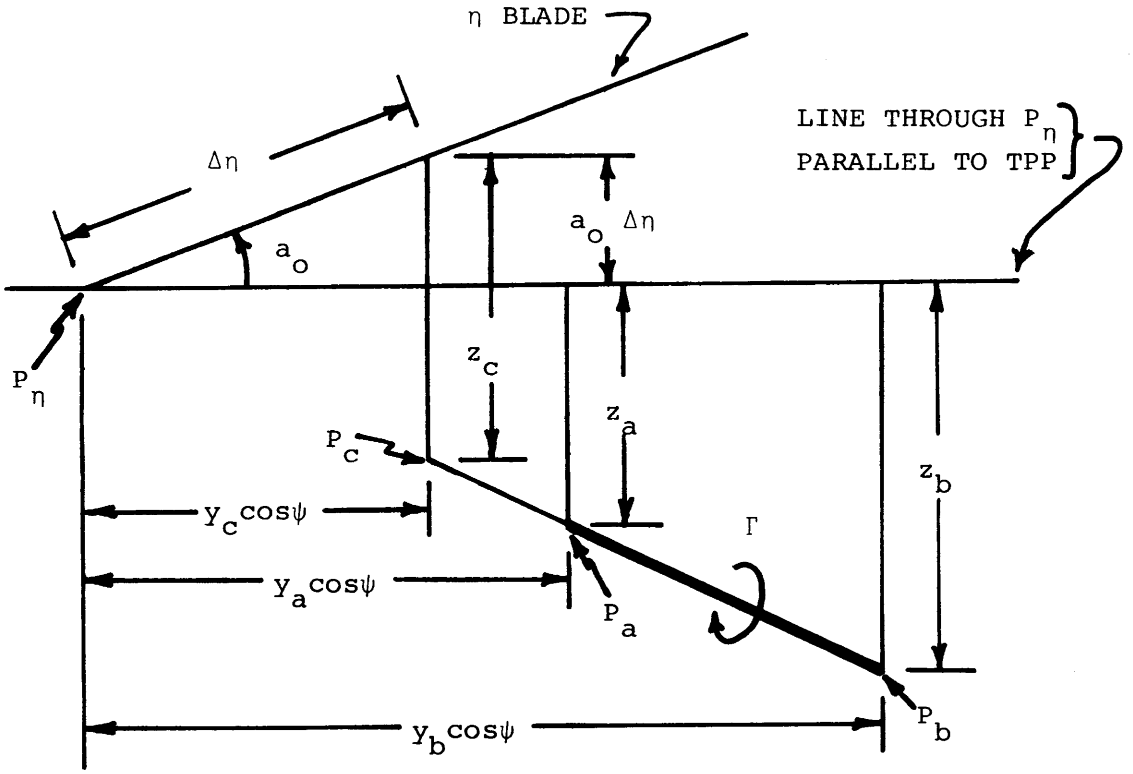

FIG. 18 SIDE VIEW OF BLADE-VORTEX LINE SEGMENT INTERACTION GEOMETRY, LOOKING PARALLEL TO TPP AND PERPENDICULAR TO ROTOR BLADE

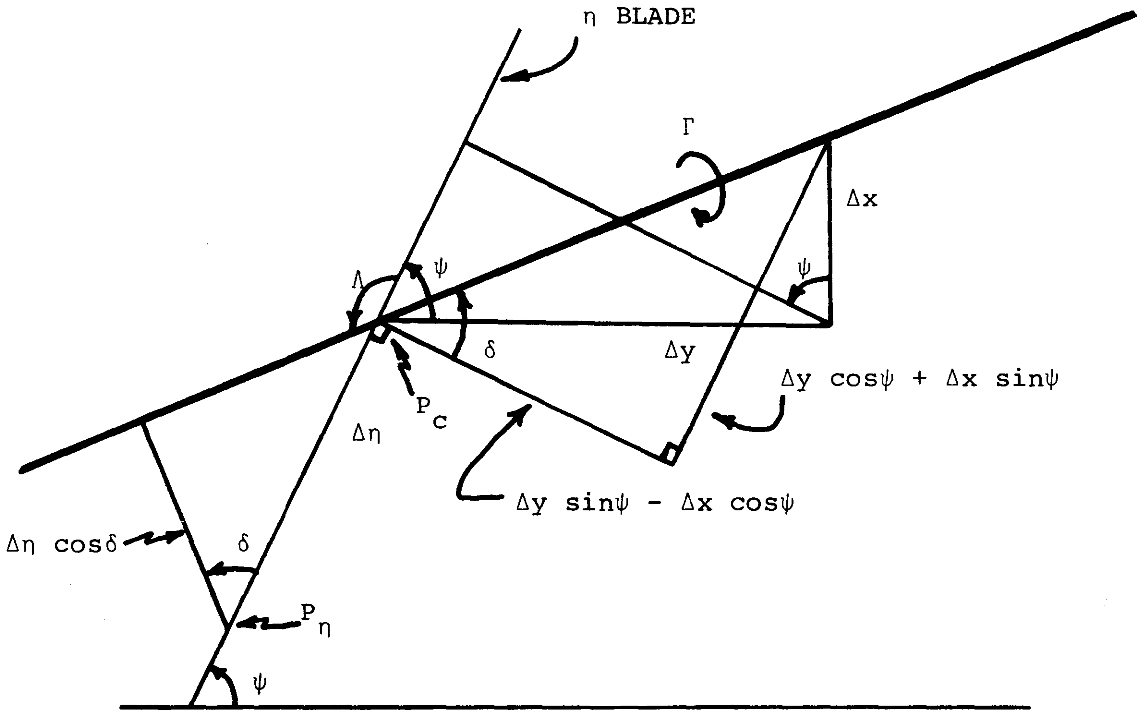

FIG. 19 TOP (TPP) VIEW OF BLADE-VORTEX LINE GEOMETRY FOR LIFTING-SURFACE THEORY

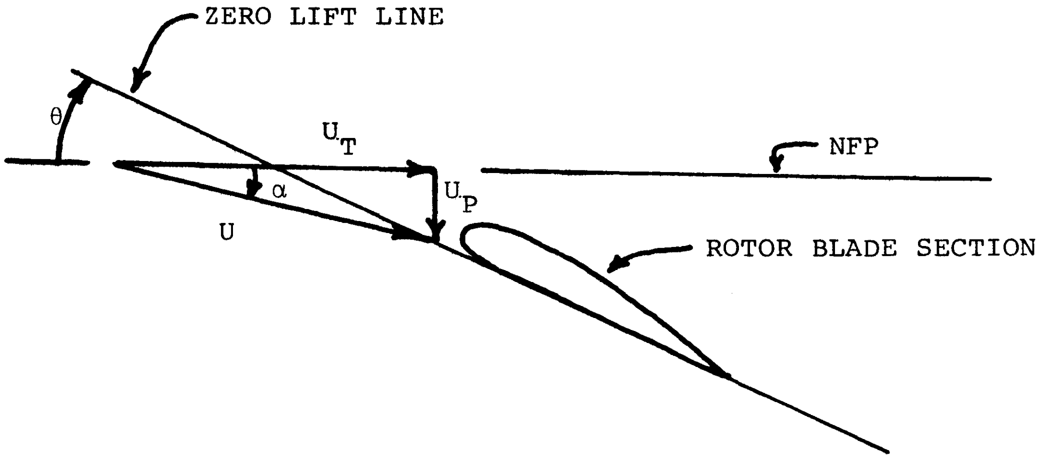

FIG. 20 ROTOR-BLADE SECTION ANGLE OF ATTACK AND PITCH ANGLE

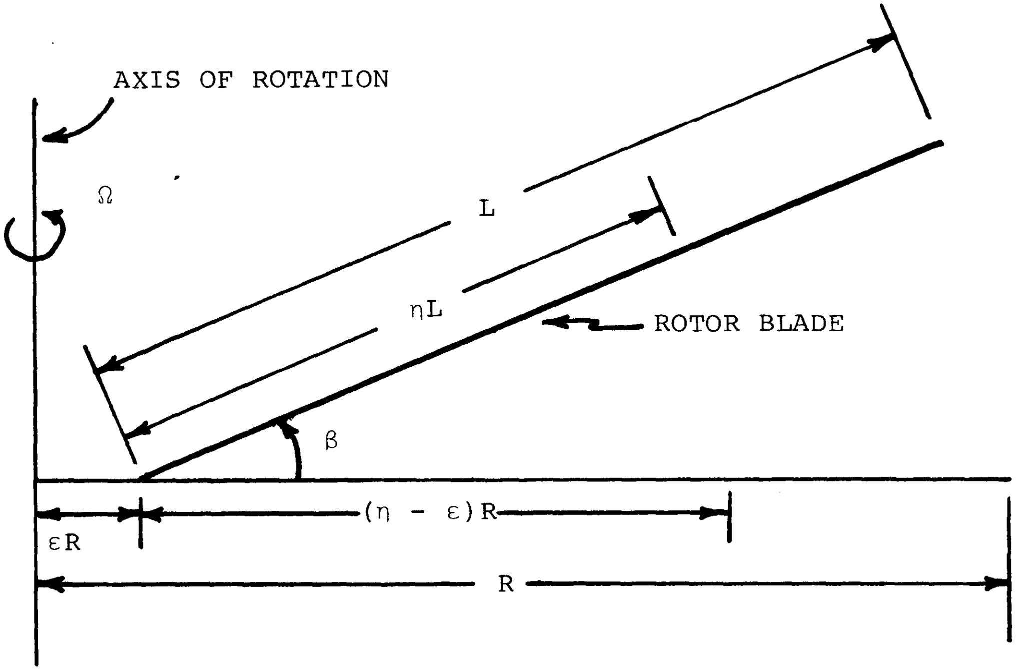

FIG. 21 ROTOR-BLADE FLAPPING HINGE OFFSET, WHERE $(\eta - \varepsilon)R = \bar{\eta}L$

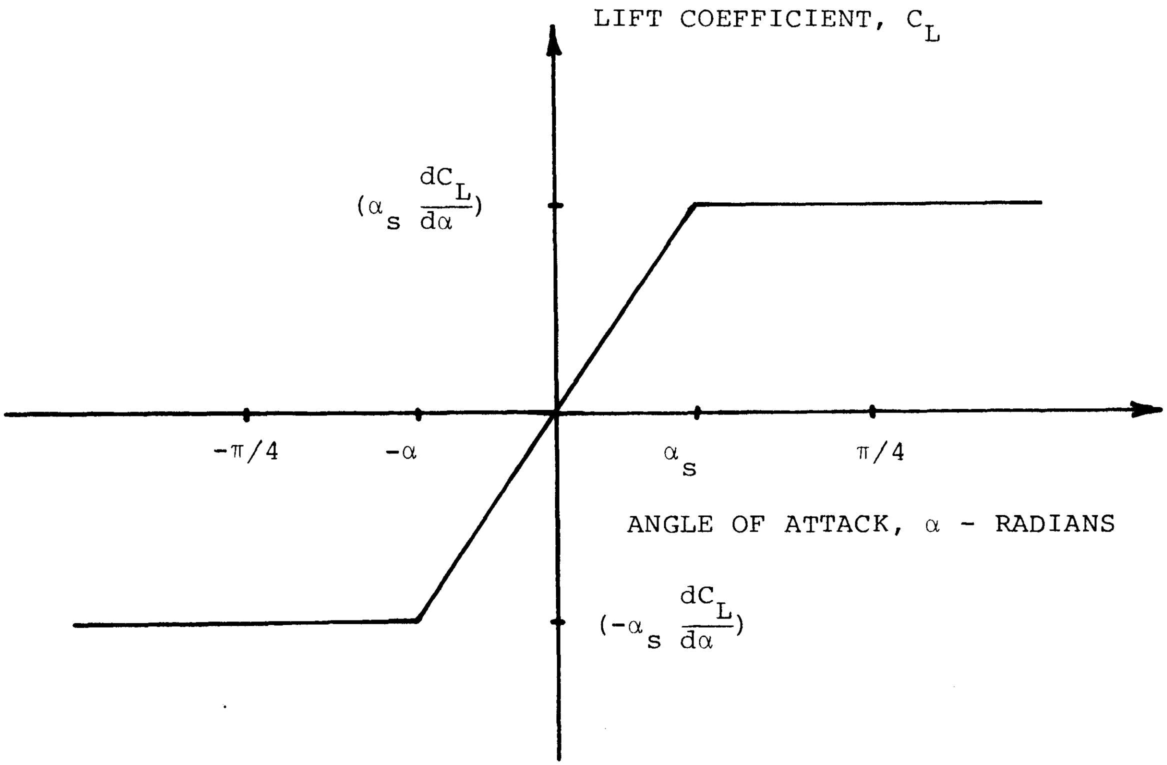

FIG. 22 C_L VS. α FOR NUMERICAL SOLUTION TO BLADE MOTION

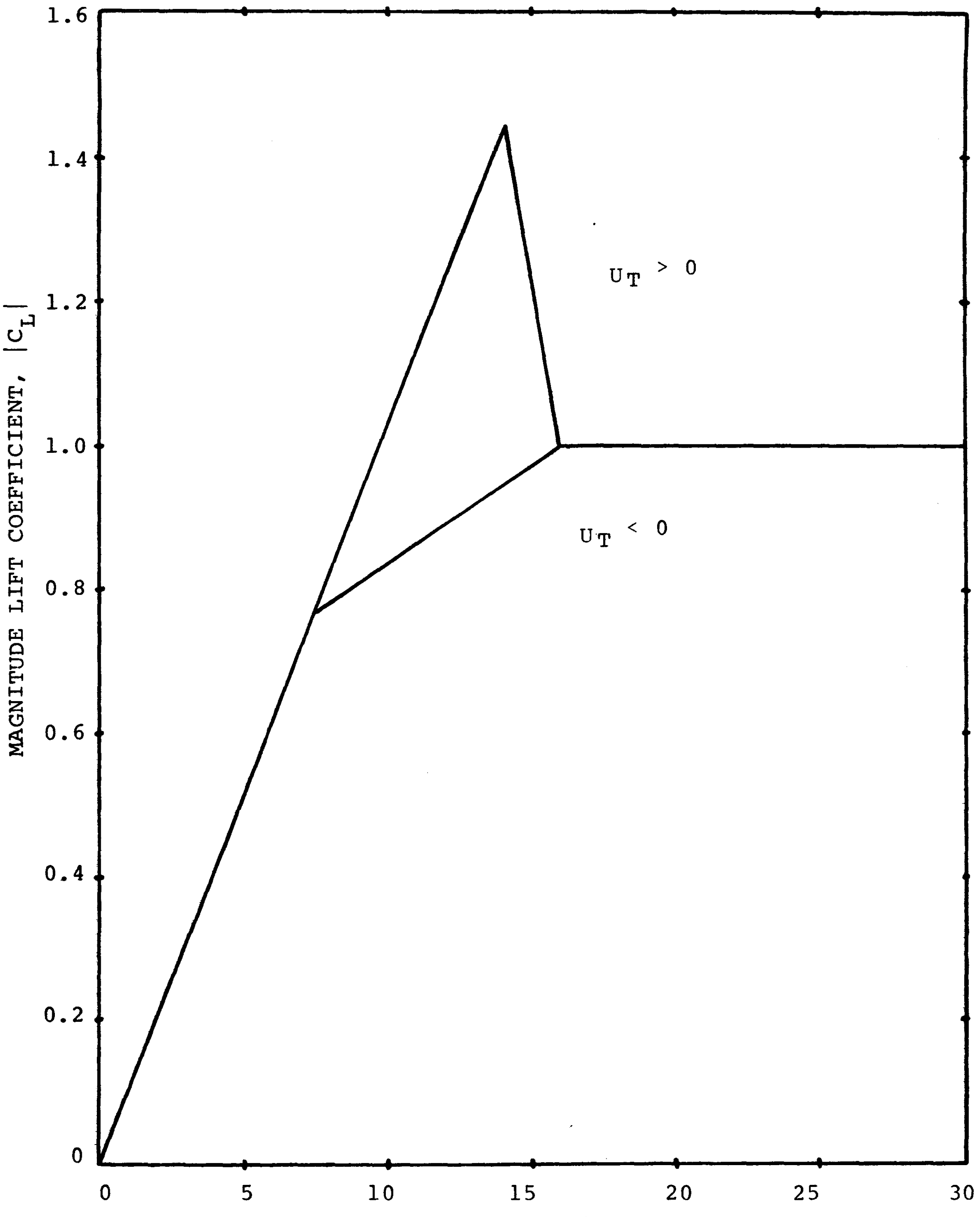

FIG. 23a STATIC C_L VS. α FOR M = 0.2

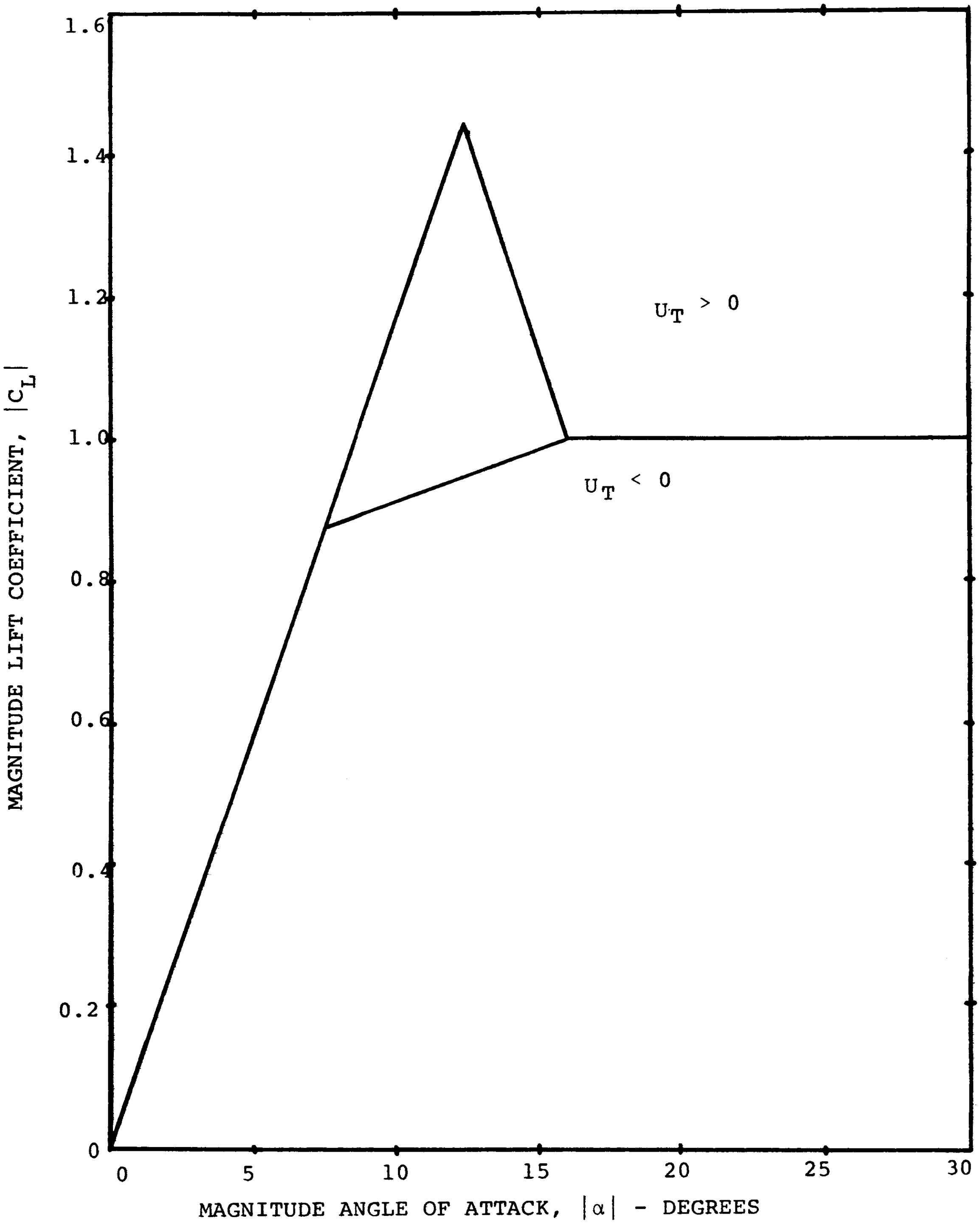

FIG. 23b STATIC C_L VS α FOR $M = 0.5$

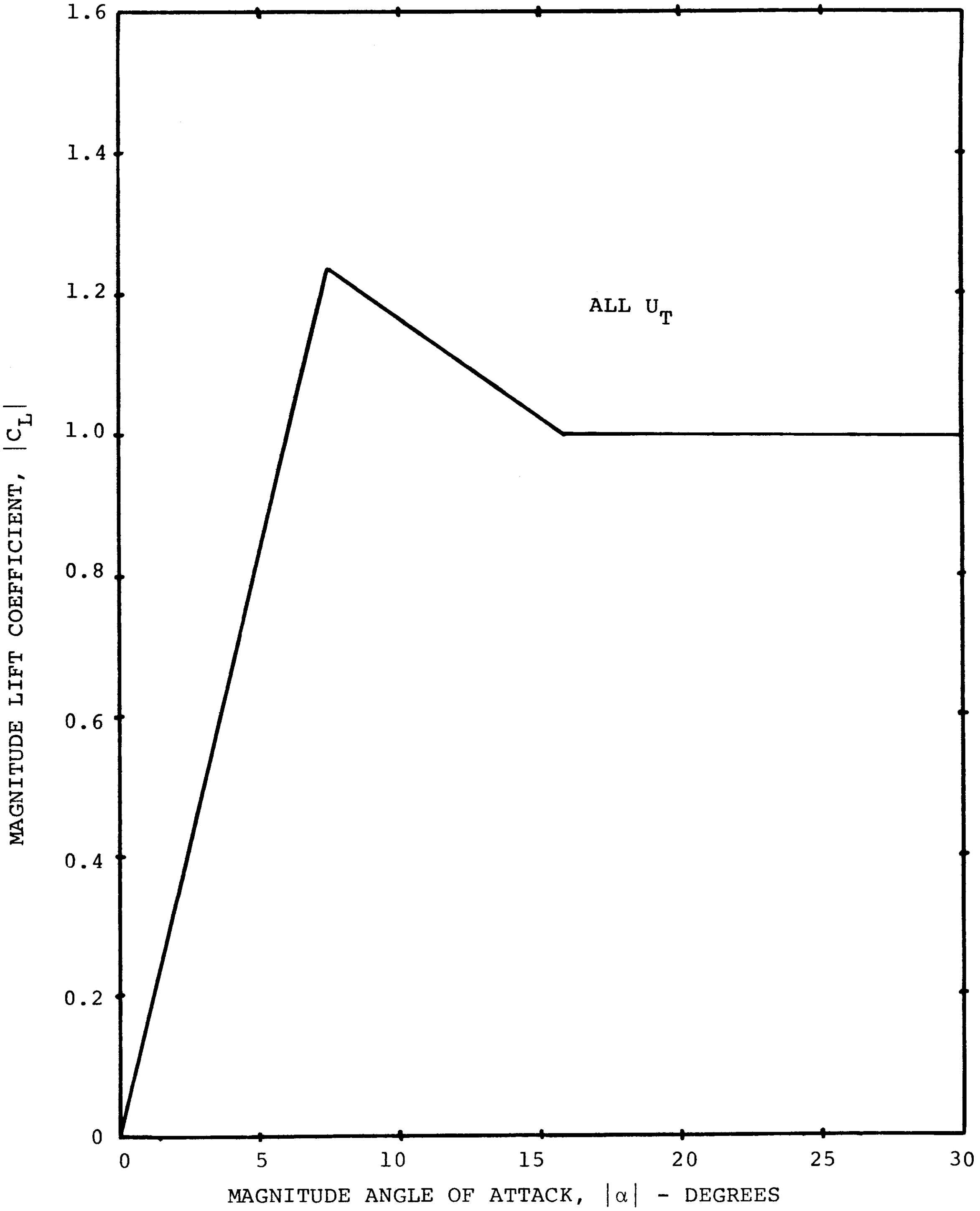

FIG. 23c STATIC C_L VS α FOR $M = 0.9$

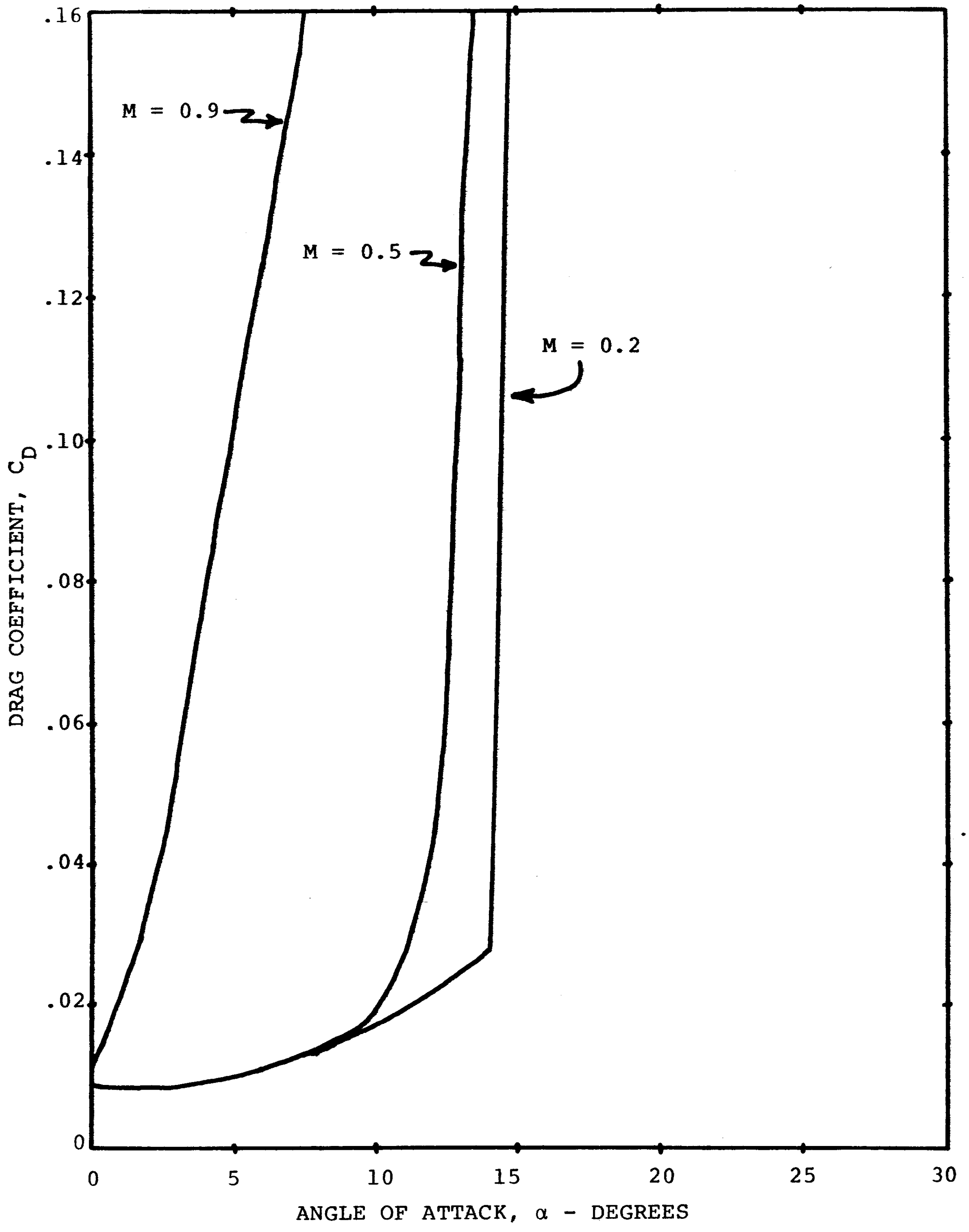

FIG. 24 STATIC C_D VS. α

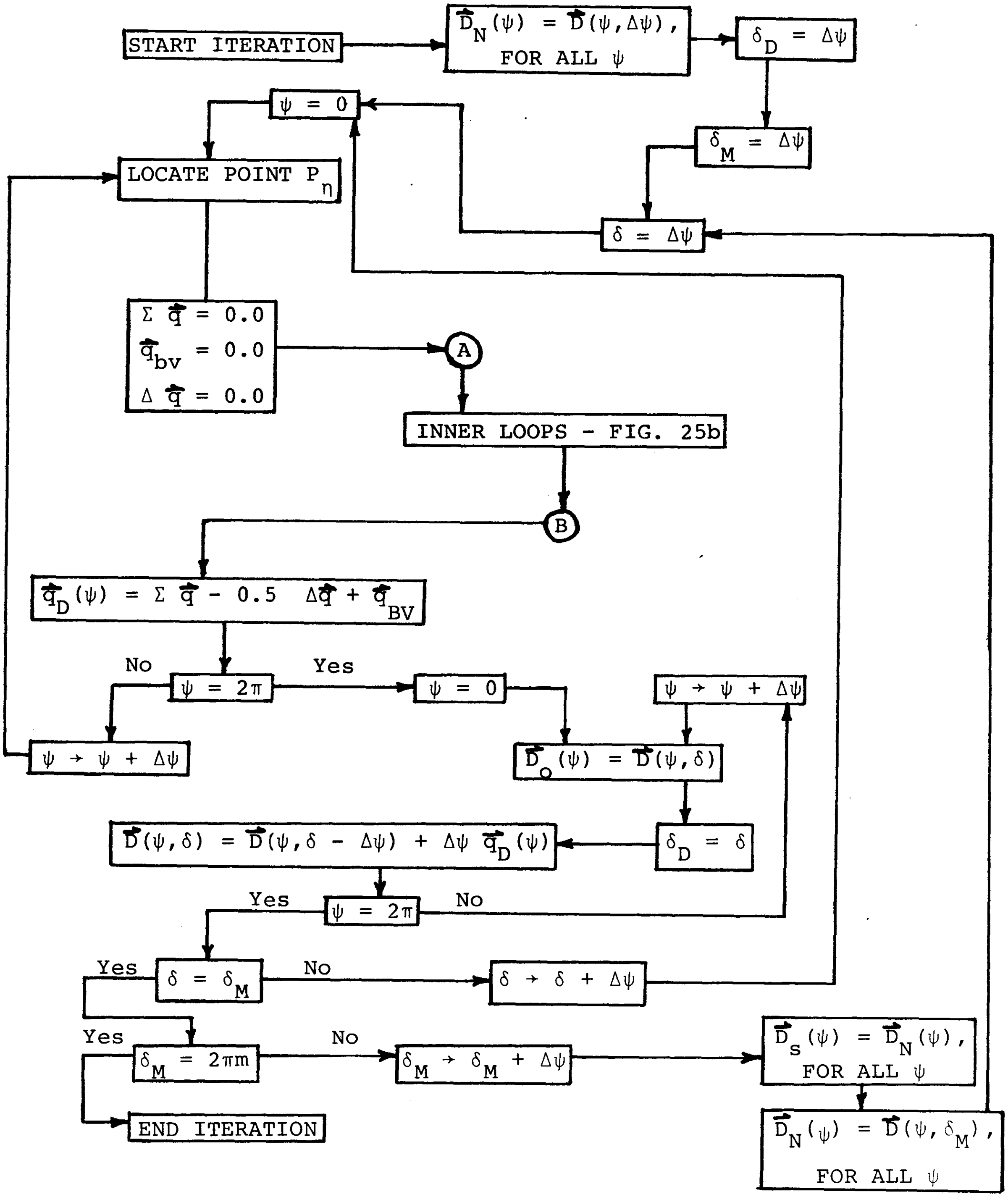

FIG. 25a BASIC DISTORTION ITERATION - OUTER LOOPS

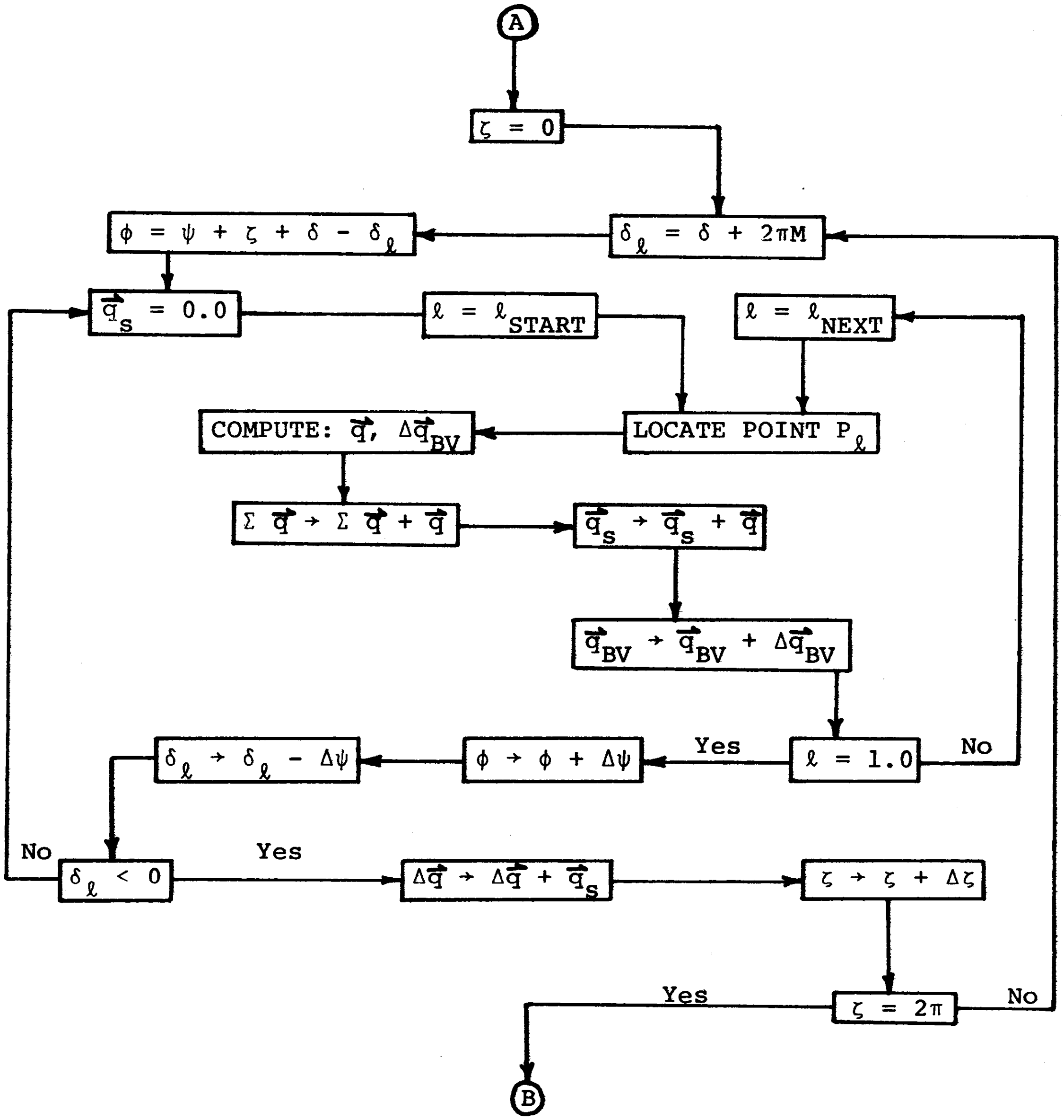

FIG. 25b BASIC DISTORTION ITERATION - INNER LOOPS

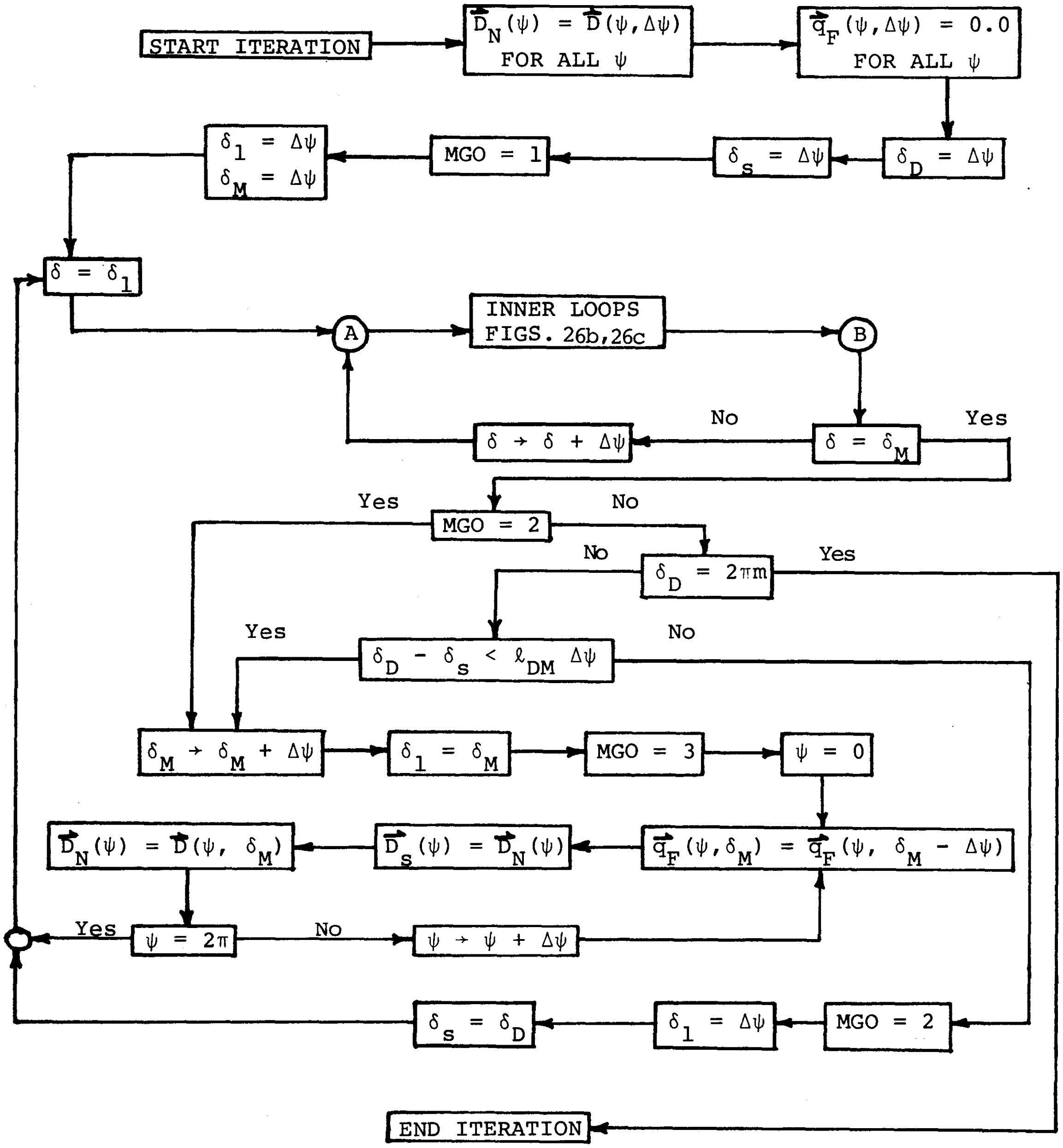

FIG. 26a ACTUAL DISTORTION ITERATION

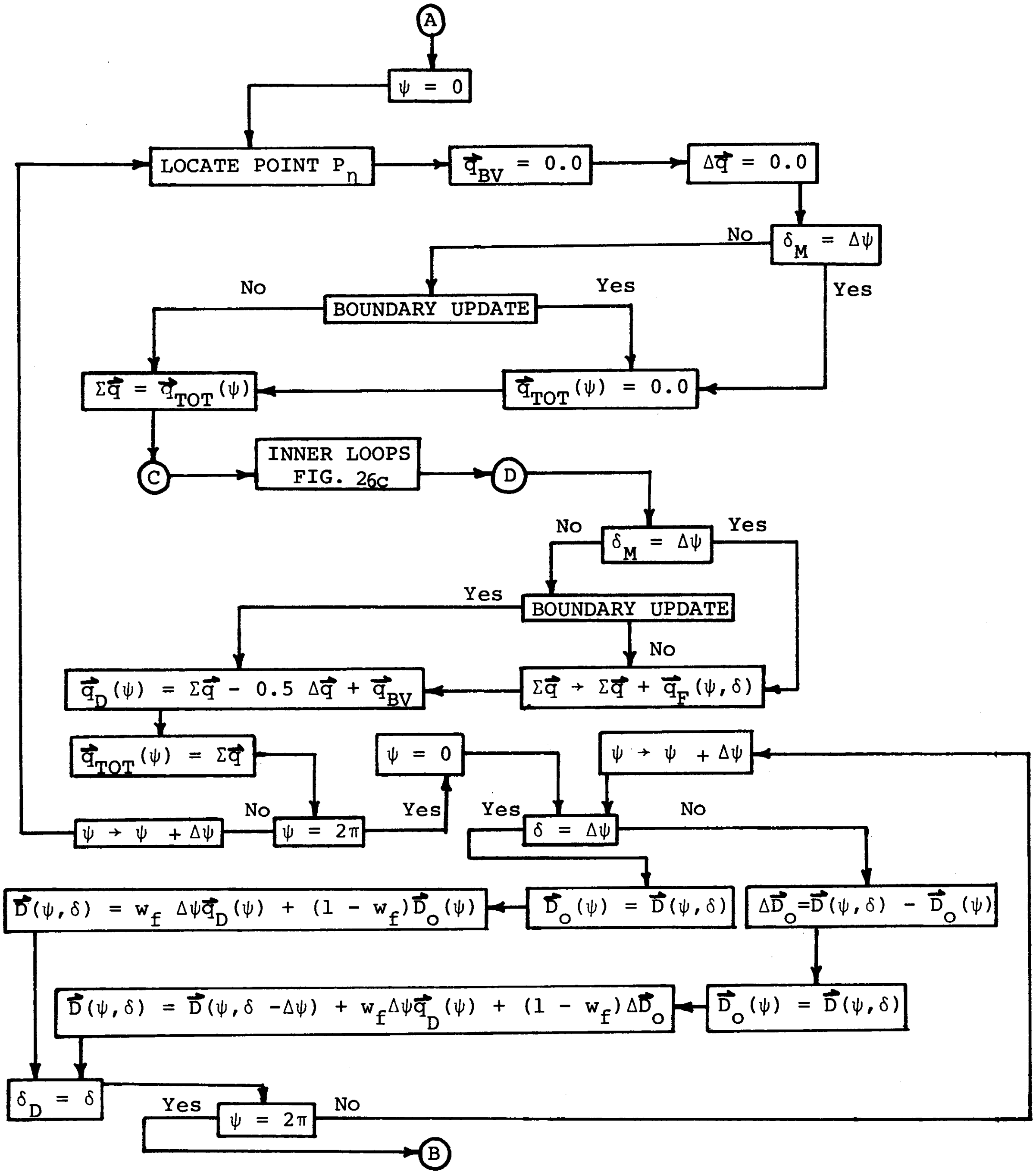

FIG. 26b ACTUAL DISTORTION ITERATION

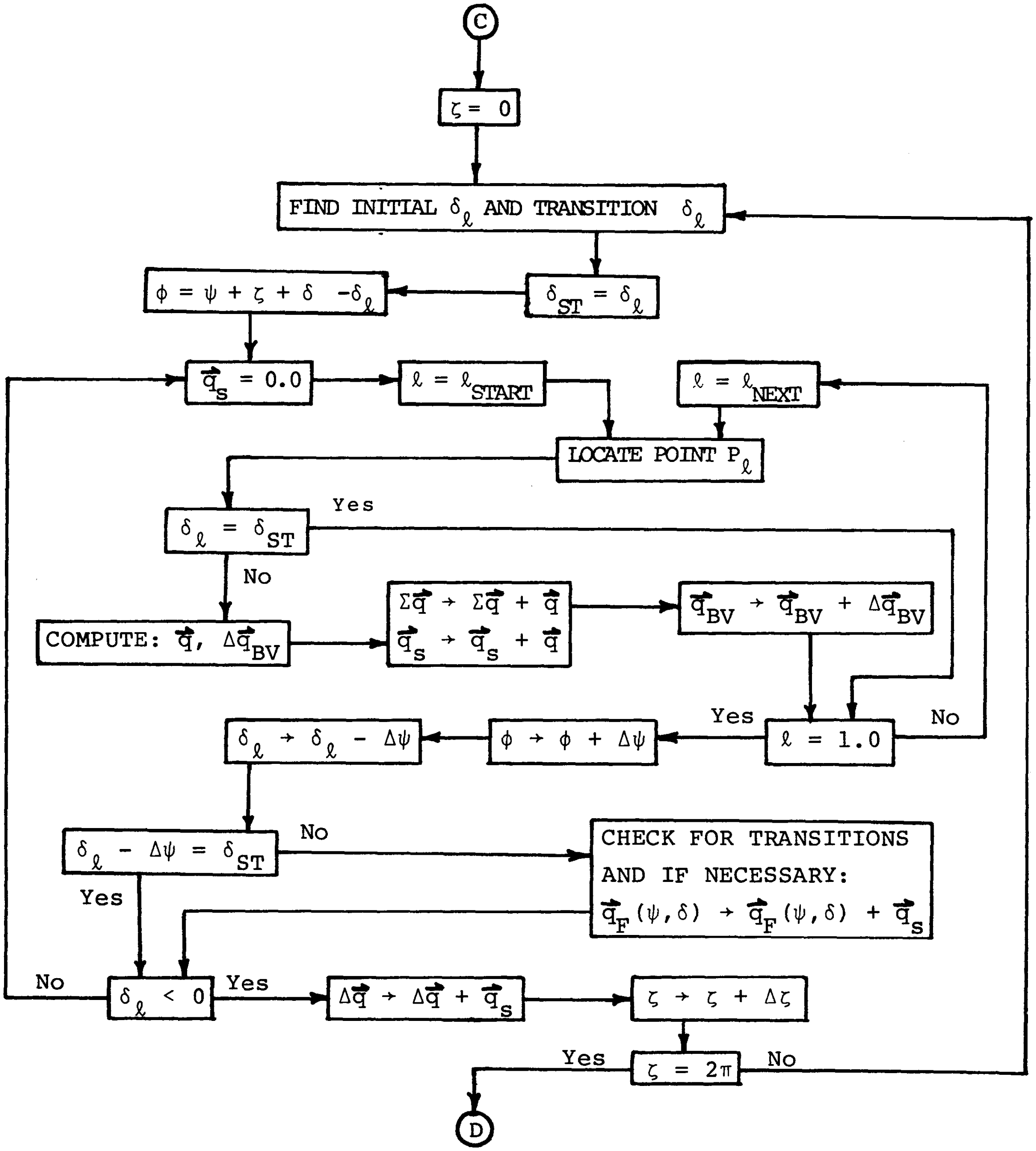

FIG. 26c ACTUAL DISTORTION ITERATION

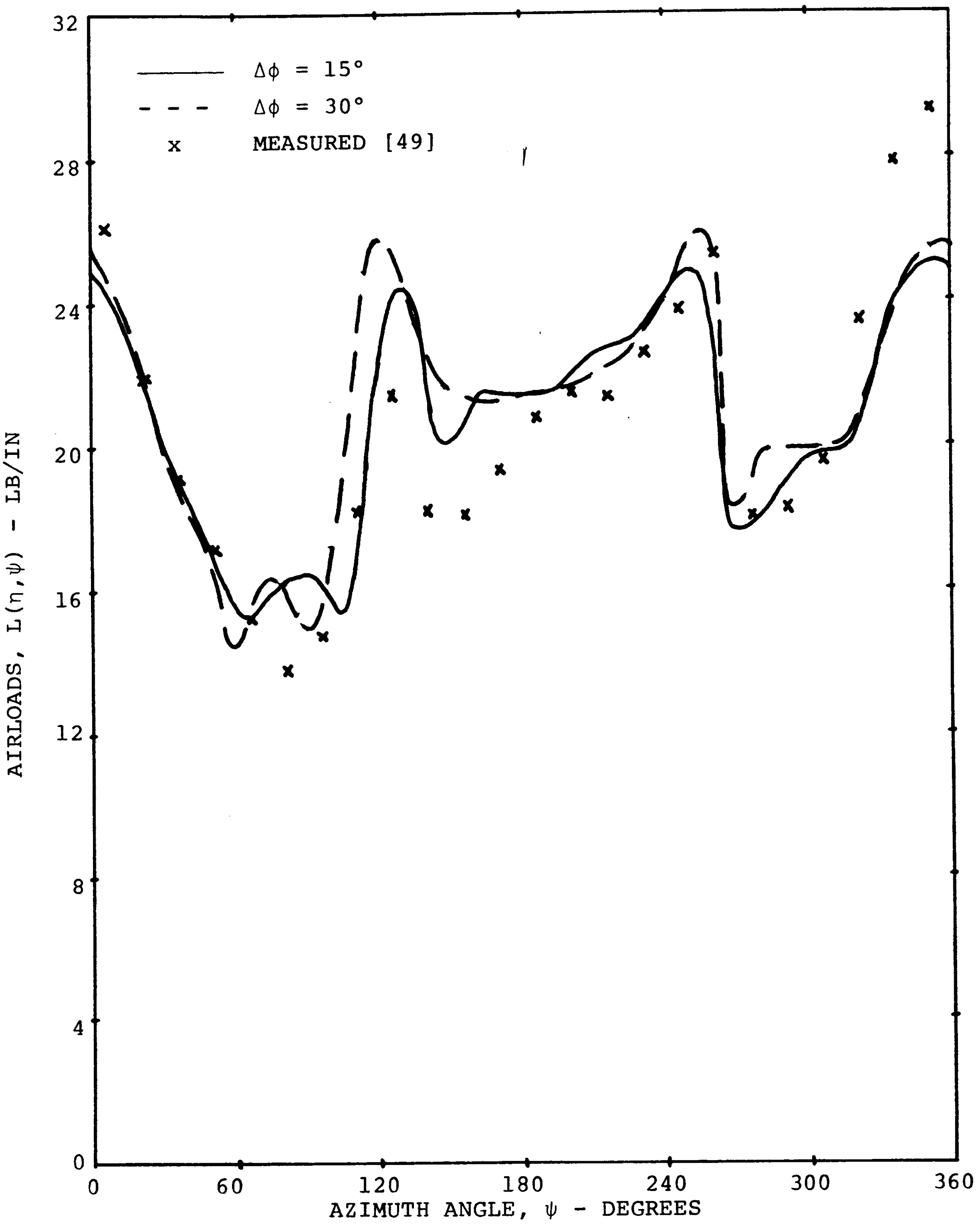

FIG. 27 RIGID WAKE AIRLOADS, $\Delta\phi$ = 15° VS $\Delta\phi$ = 30°, 85% RADIUS, 4 BLADE ROTOR, μ = 0.18

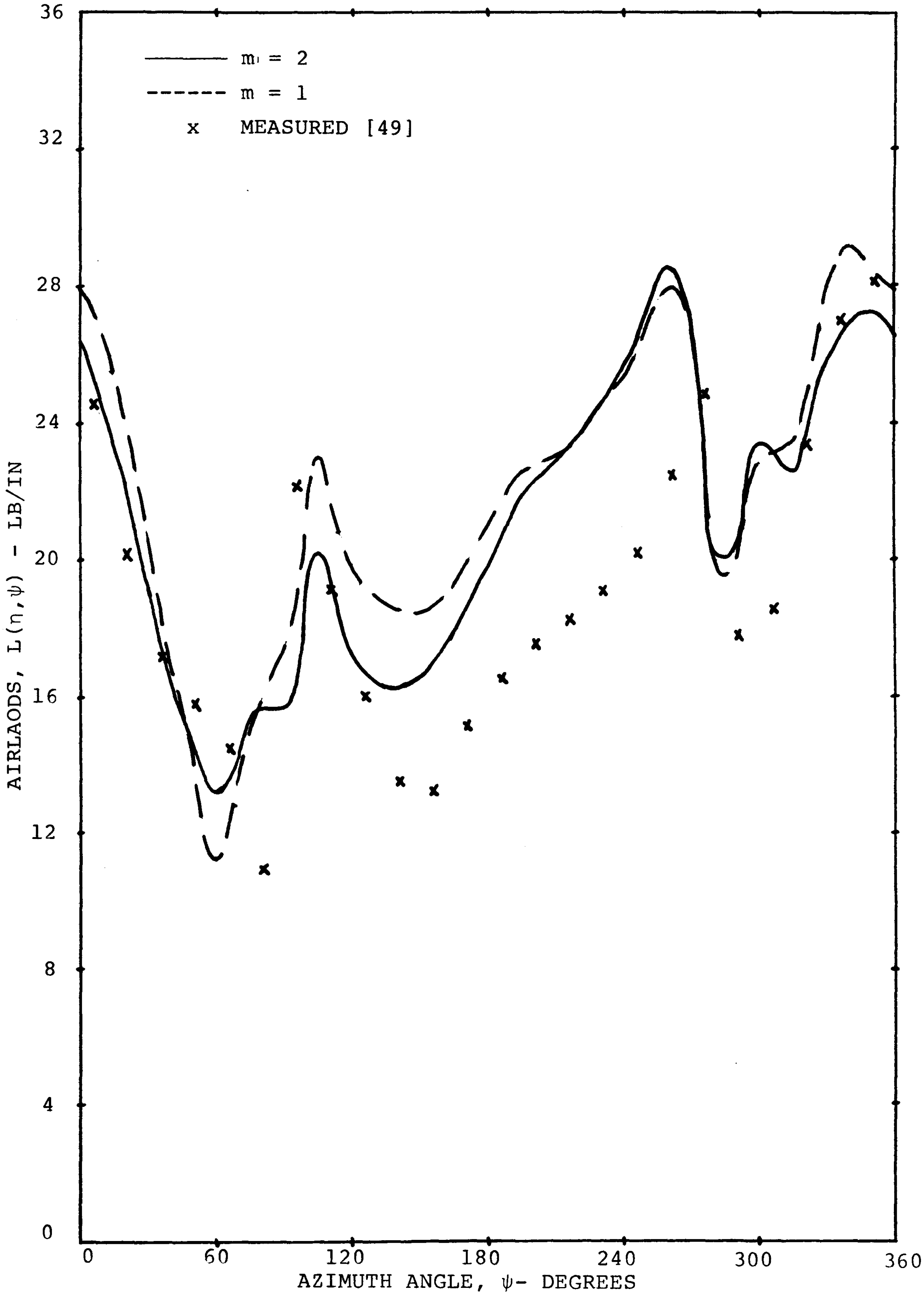

FIG. 28 RIGID WAKE AIRLOADS, m = 2 VS. m = 1, 95% RADIUS, 4 BLADE ROTOR, μ = 0.18

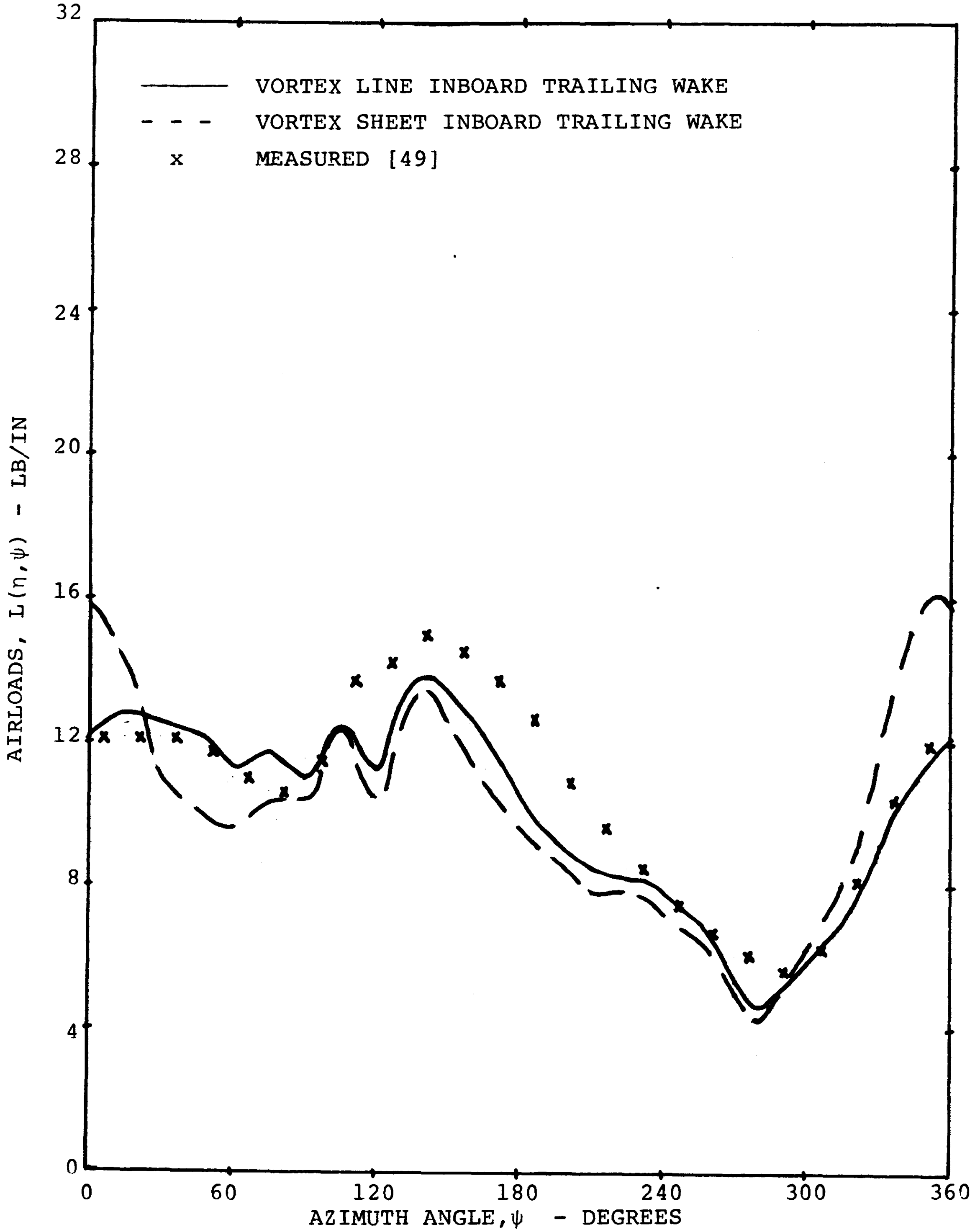

FIG. 29 RIGID WAKE AIRLOADS, VORTEX LINE VS. VORTEX SHEET INBOARD TRAILING WAKE, 55% RADIUS, 4 BLADE ROTOR, $\mu = 0.18$

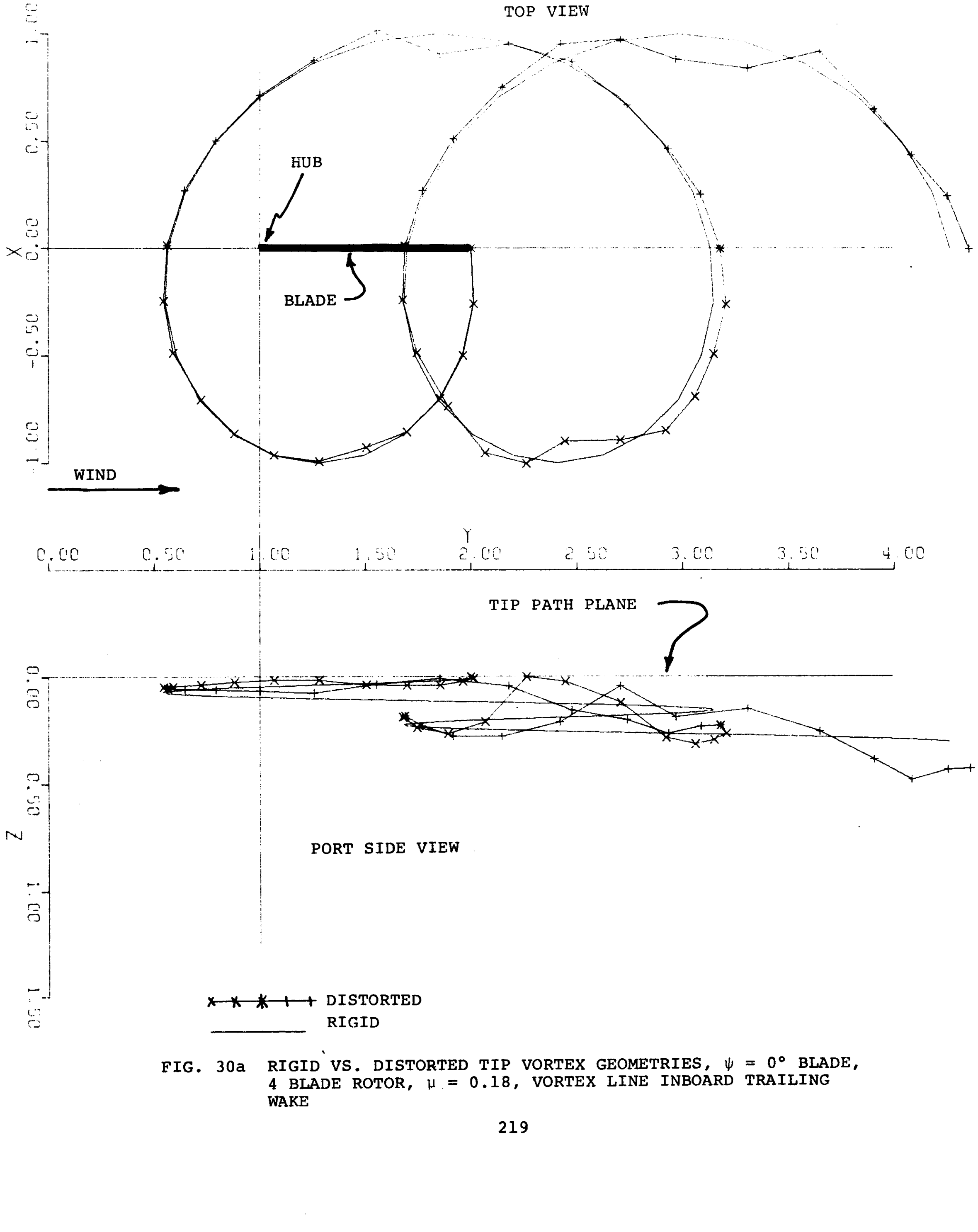

FIG. 30a RIGID VS. DISTORTED TIP VORTEX GEOMETRIES, $\psi = 0°$ BLADE, 4 BLADE ROTOR, $\mu = 0.18$, VORTEX LINE INBOARD TRAILING WAKE

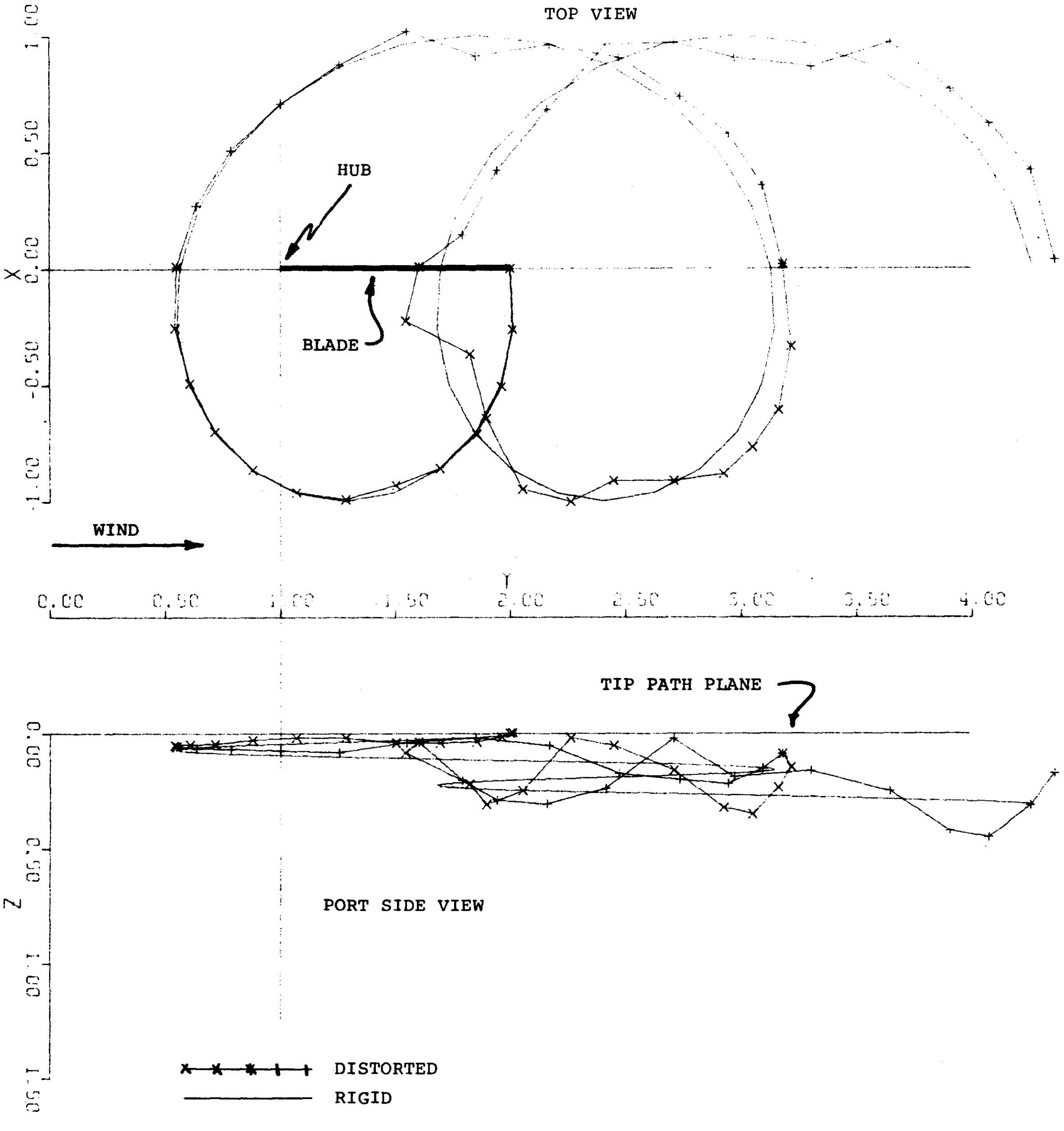

FIG. 30b RIGID VS. DISTORTED TIP VORTEX GEOMETRIES, ψ = 0° BLADE, 4 BLADE ROTOR, $\mu = 0.18$, VORTEX SHEET INBOARD TRAILING WAKE

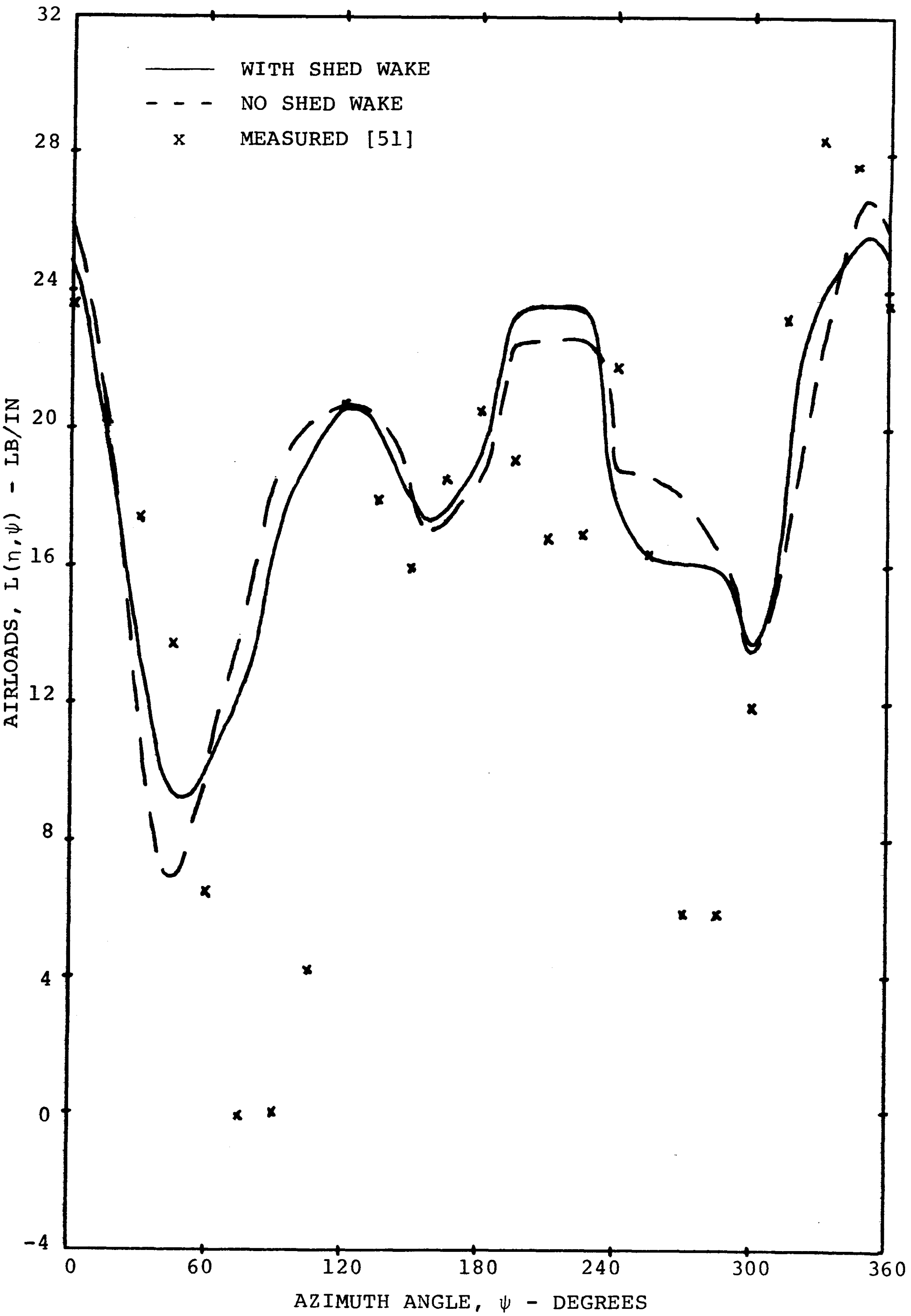

FIG. 31 RIGID WAKE AIRLOADS, EFFECT OF SHED WAKE, 85% RADIUS, 6-BLADE ROTOR, $\mu = 0.108$

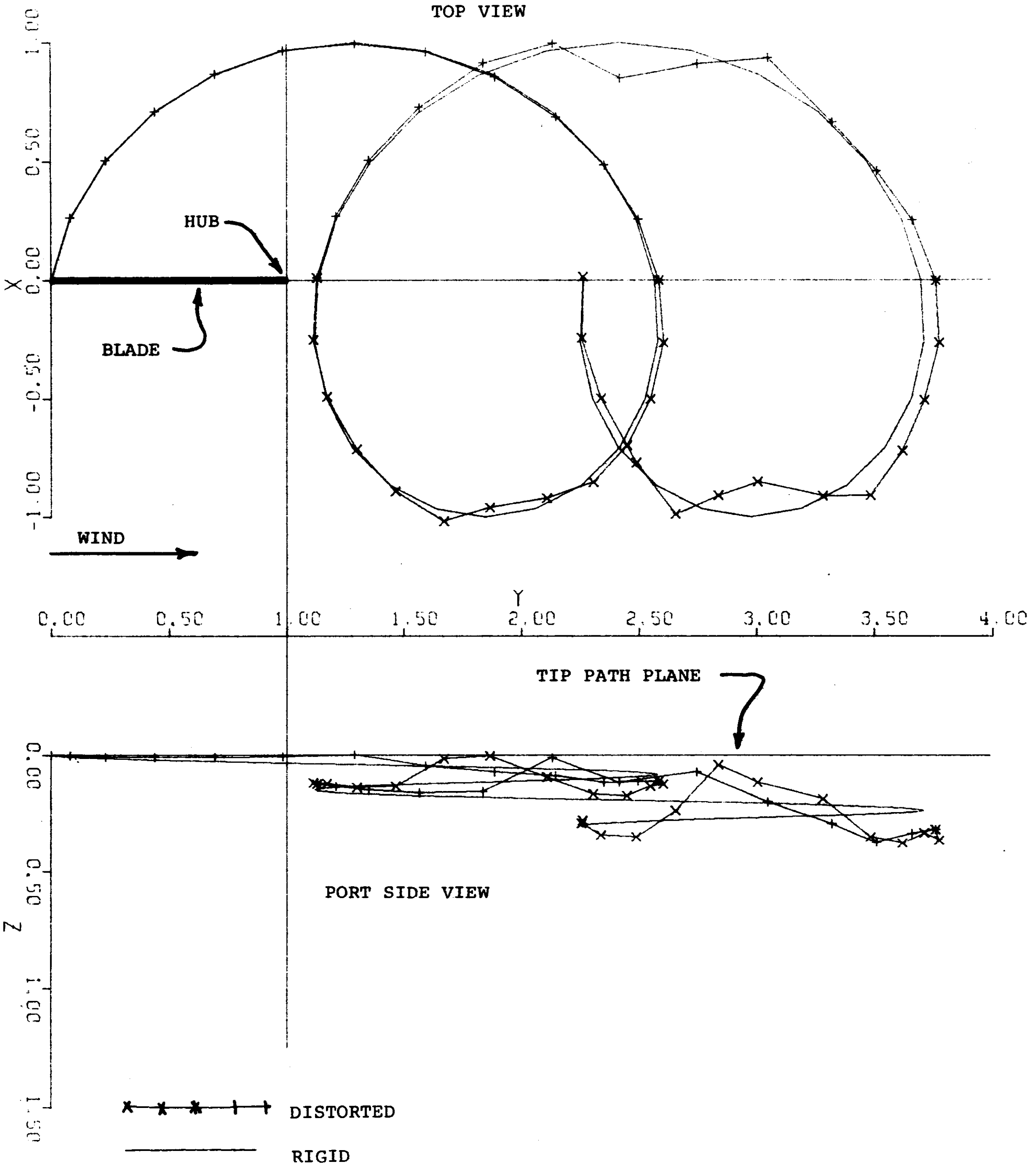

FIG. 32 RIGID VS. DISTORTED TIP VORTEX GEOMETRIES, ψ = 180° BLADE, 4 BLADE ROTOR, $\mu = 0.18$, NEAR/FAR WAKE PARAMETER DM = .0005

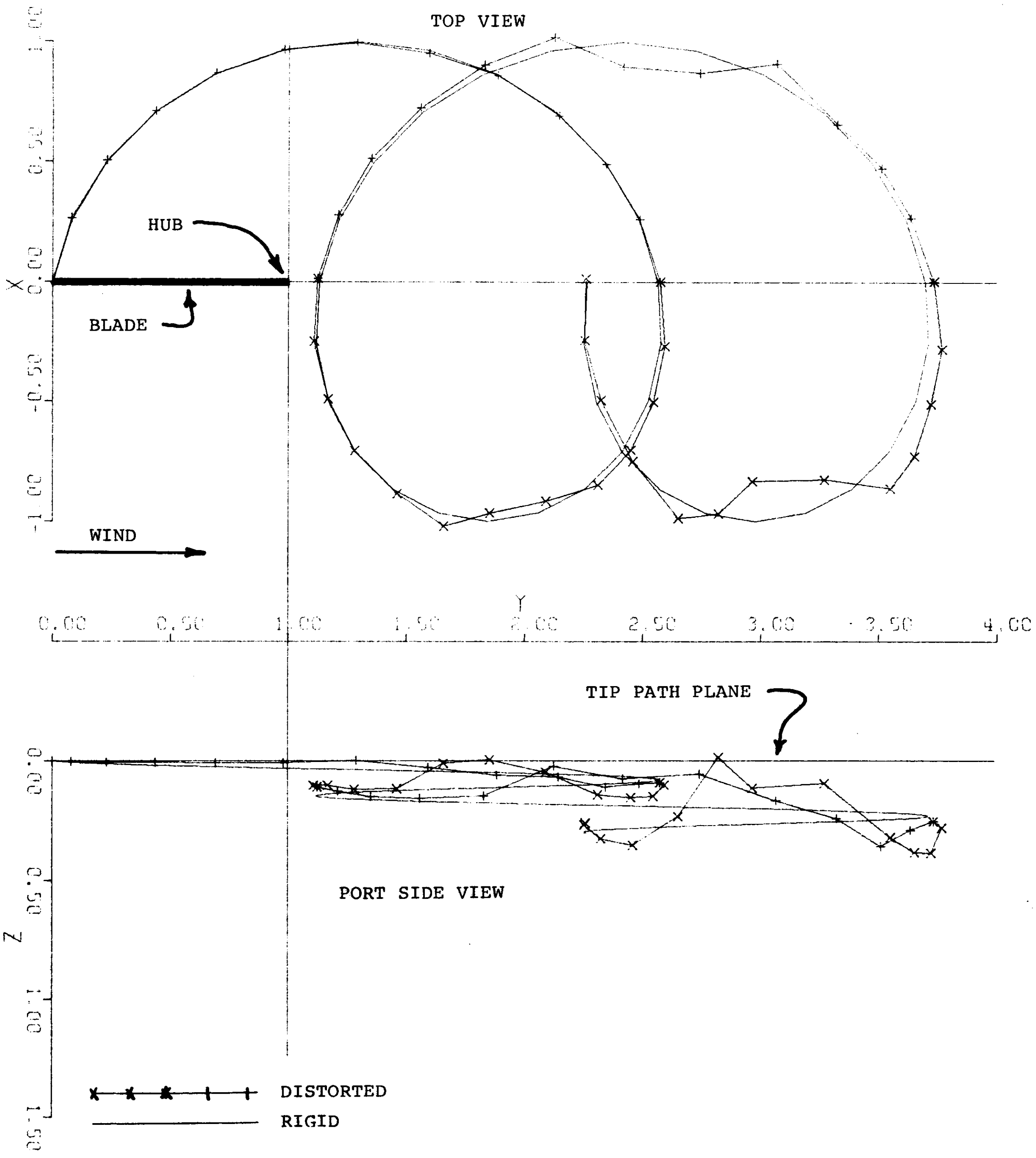

FIG. 33 RIGID VS. DISTORTED TIP VORTEX GEOMETRIES, $\psi = 180°$ BLADE, 4 BLADE ROTOR, $\mu = 0.18$, UPDATING PARAMETERS LDM = 4, NDM = 2

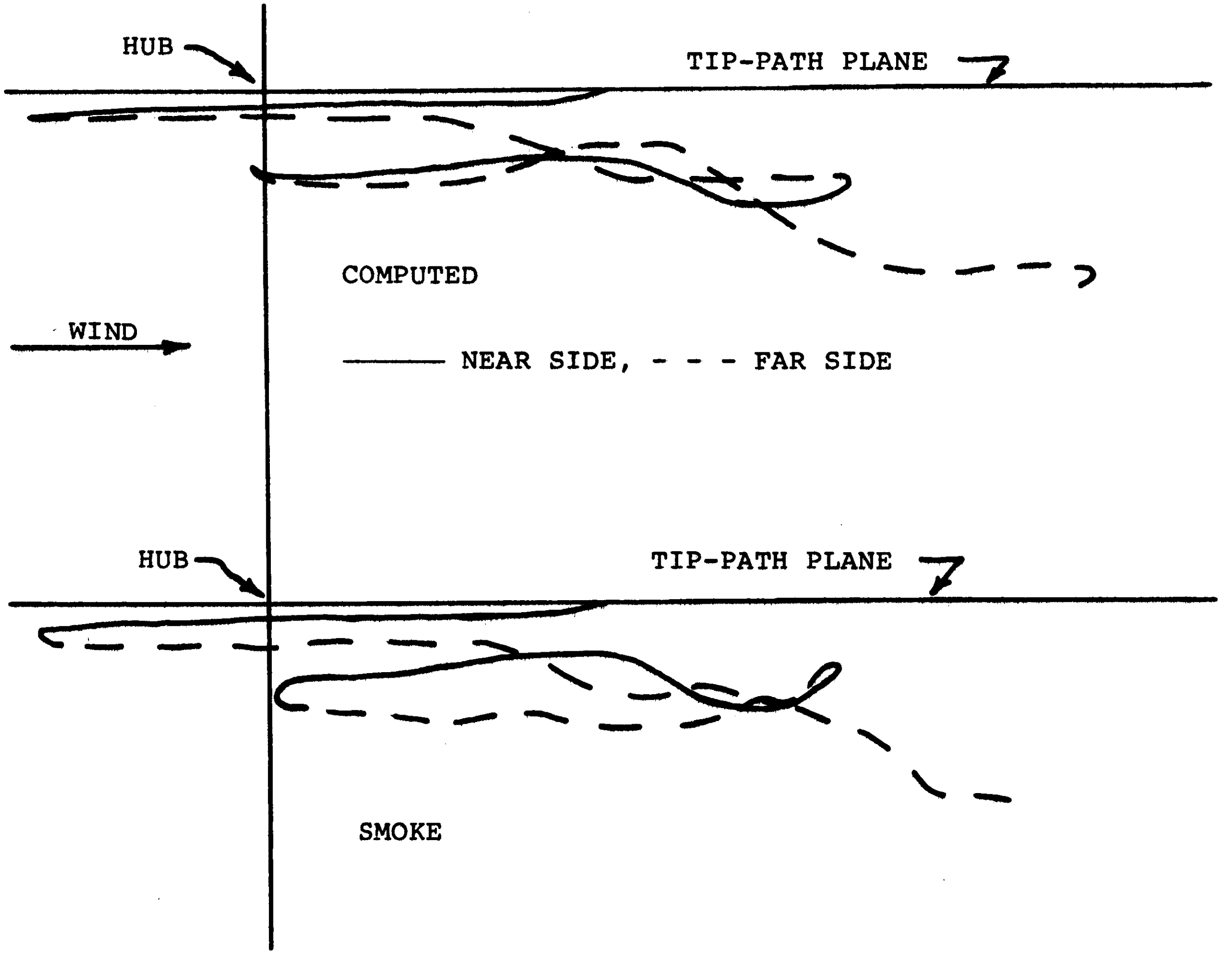

FIG. 34a COMPUTED VS. SMOKE TIP-VORTEX GEOMETRIES, $\psi = 0°$ BLADE, 1 BLADE ROTOR, $\mu = 0.10$

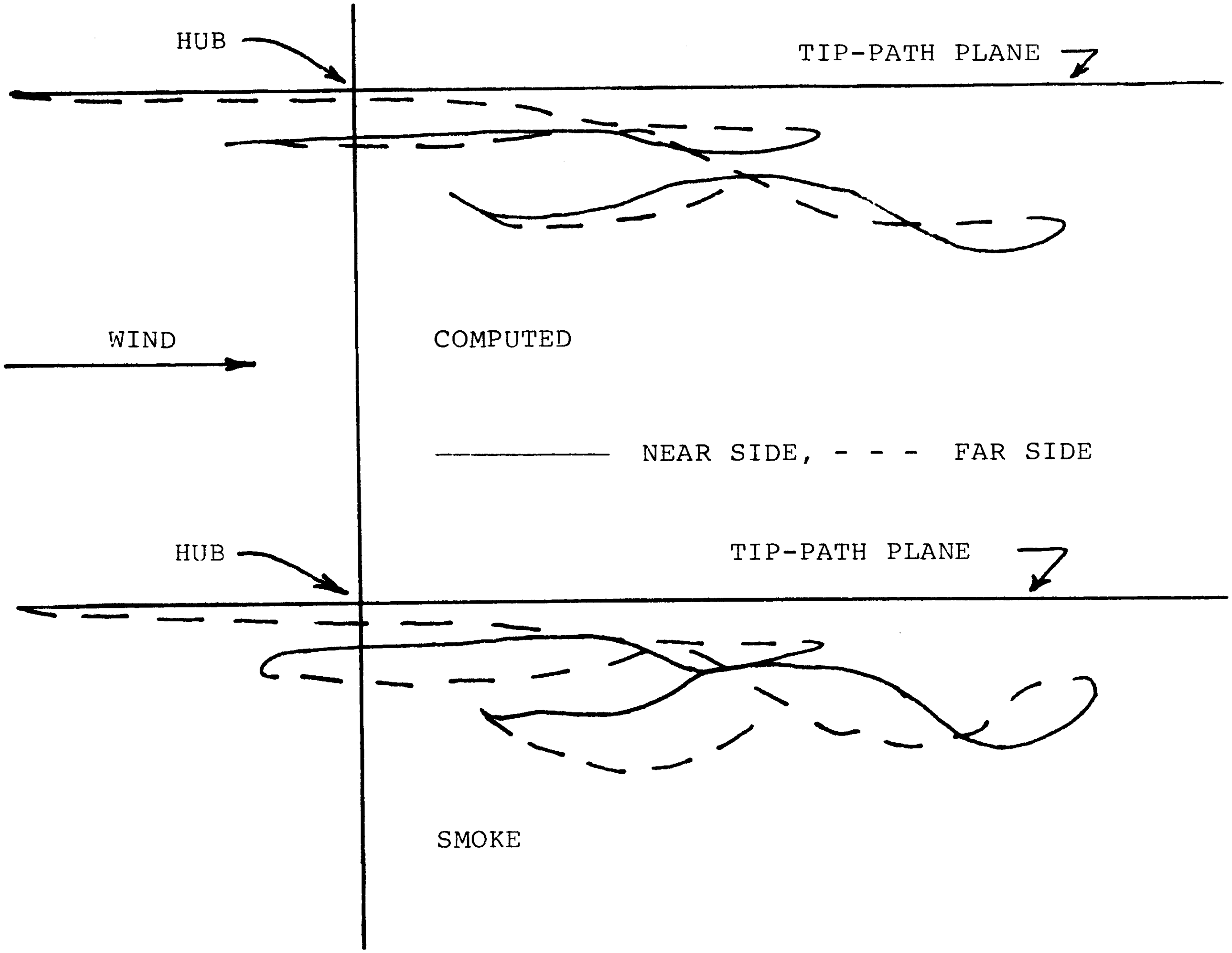

FIG. 34b COMPUTED VS. SMOKE TIP-VORTEX GEOMETRIES, ψ = 180° BLADE, 1 BLADE ROTOR, μ = 0.10

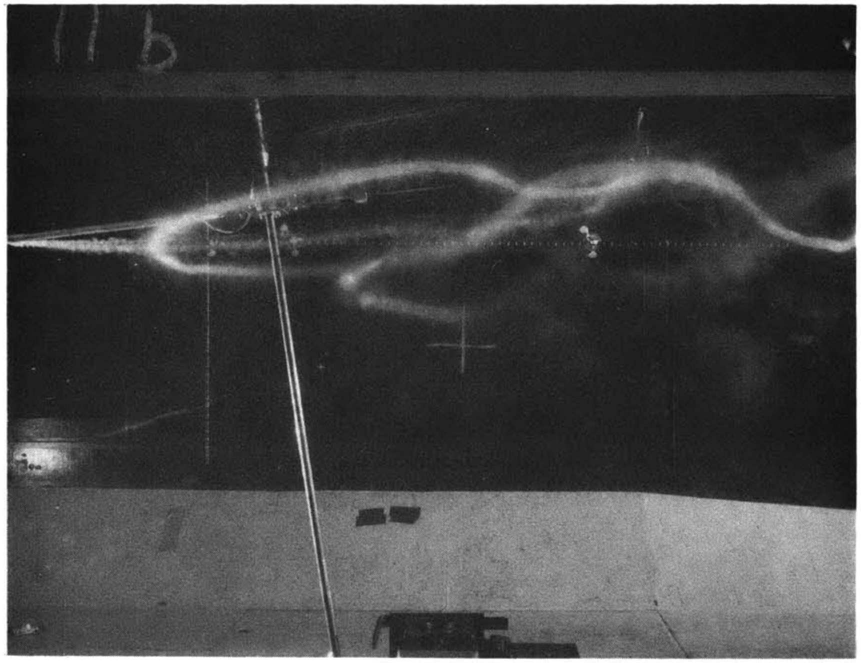

FIG. 35 SMOKE PHOTOGRAPH, ψ = 180° BLADE, 1 BLADE ROTOR $\mu = 0.10$

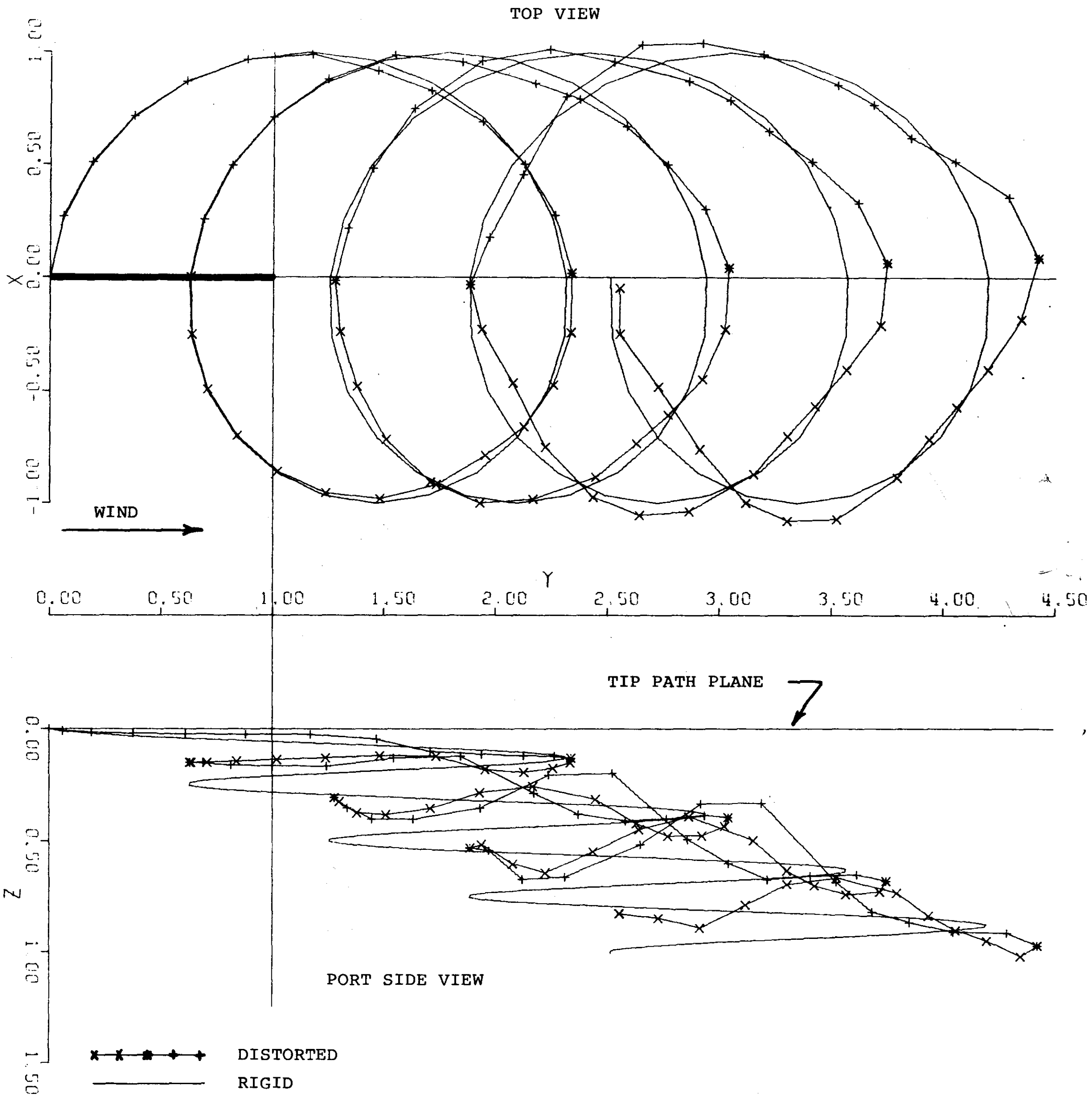

FIG. 36a RIGID VS. DISTORTED TIP VORTEX GEOMETRIES, ψ = 180° BLADE, 1-BLADE ROTOR, μ = 0.10

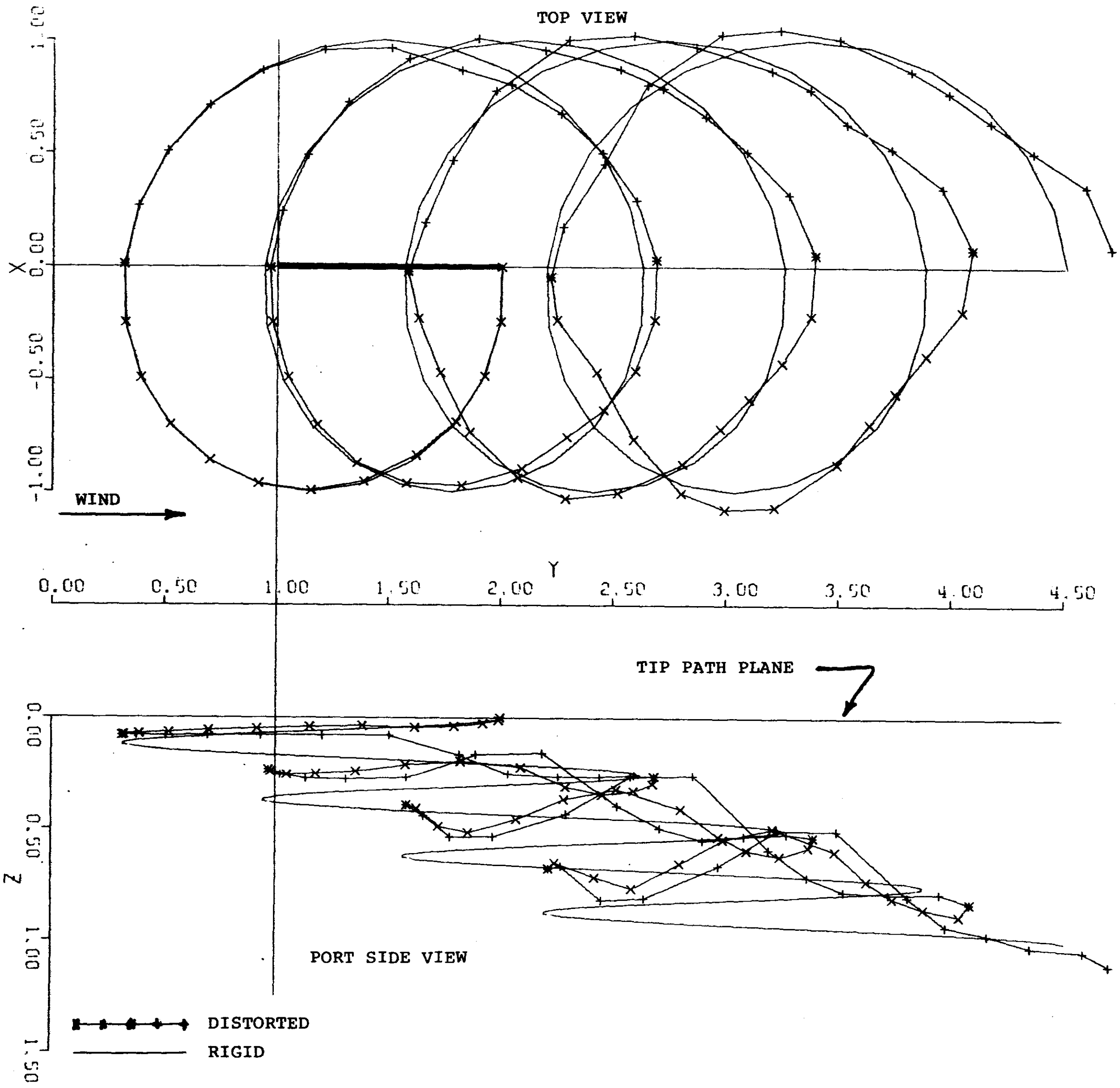

FIG. 36b RIGID VS. DISTORTED TIP VORTEX GEOMETRIES, ψ = 0° BLADE, 1-BLADE ROTOR, μ = 0.10

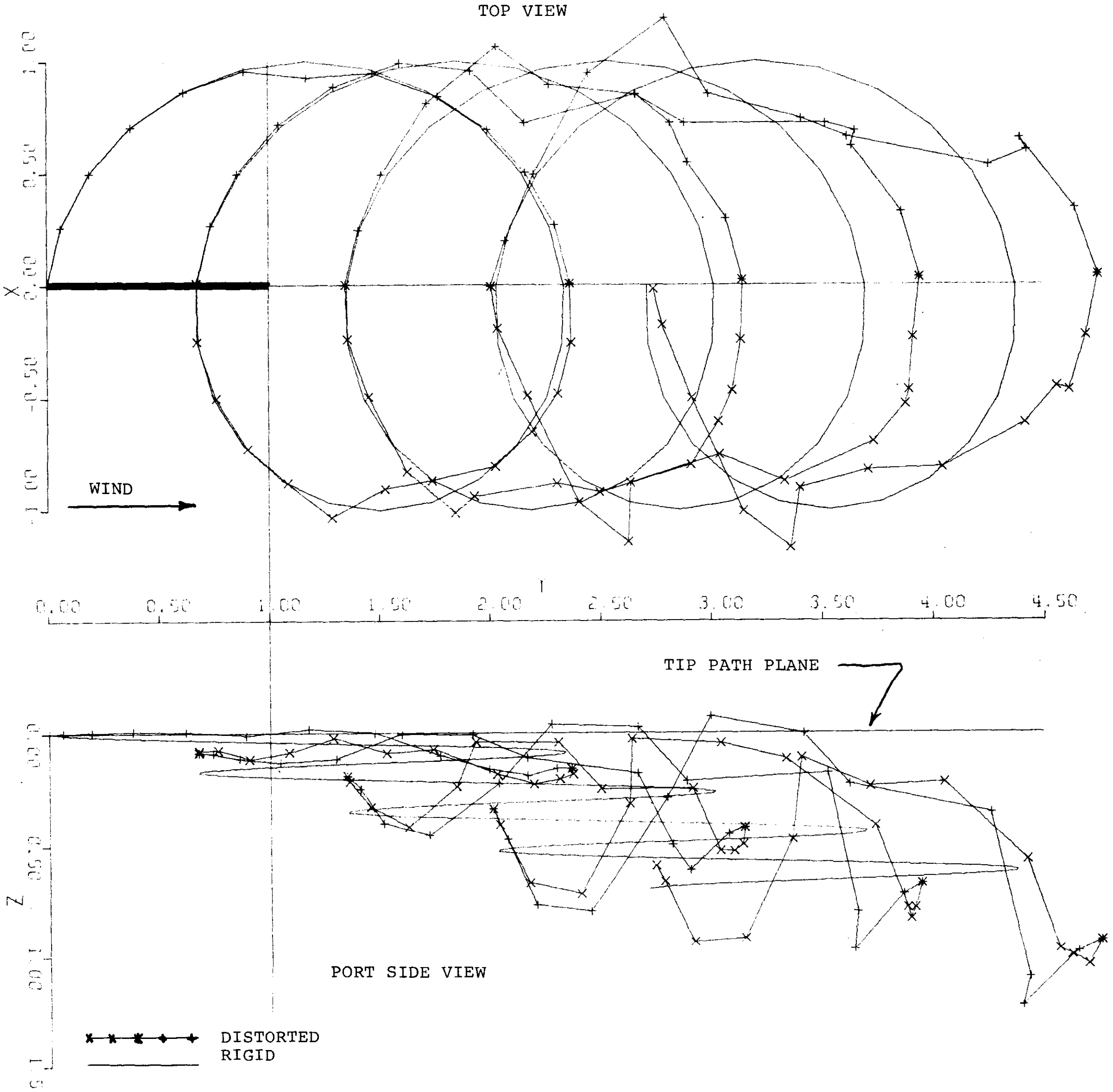

FIG. 37a RIGID VS. DISTORTED TIP VORTEX GEOMETRIES, ψ = 180° BLADE, 6-BLADE ROTOR, μ = 0.108

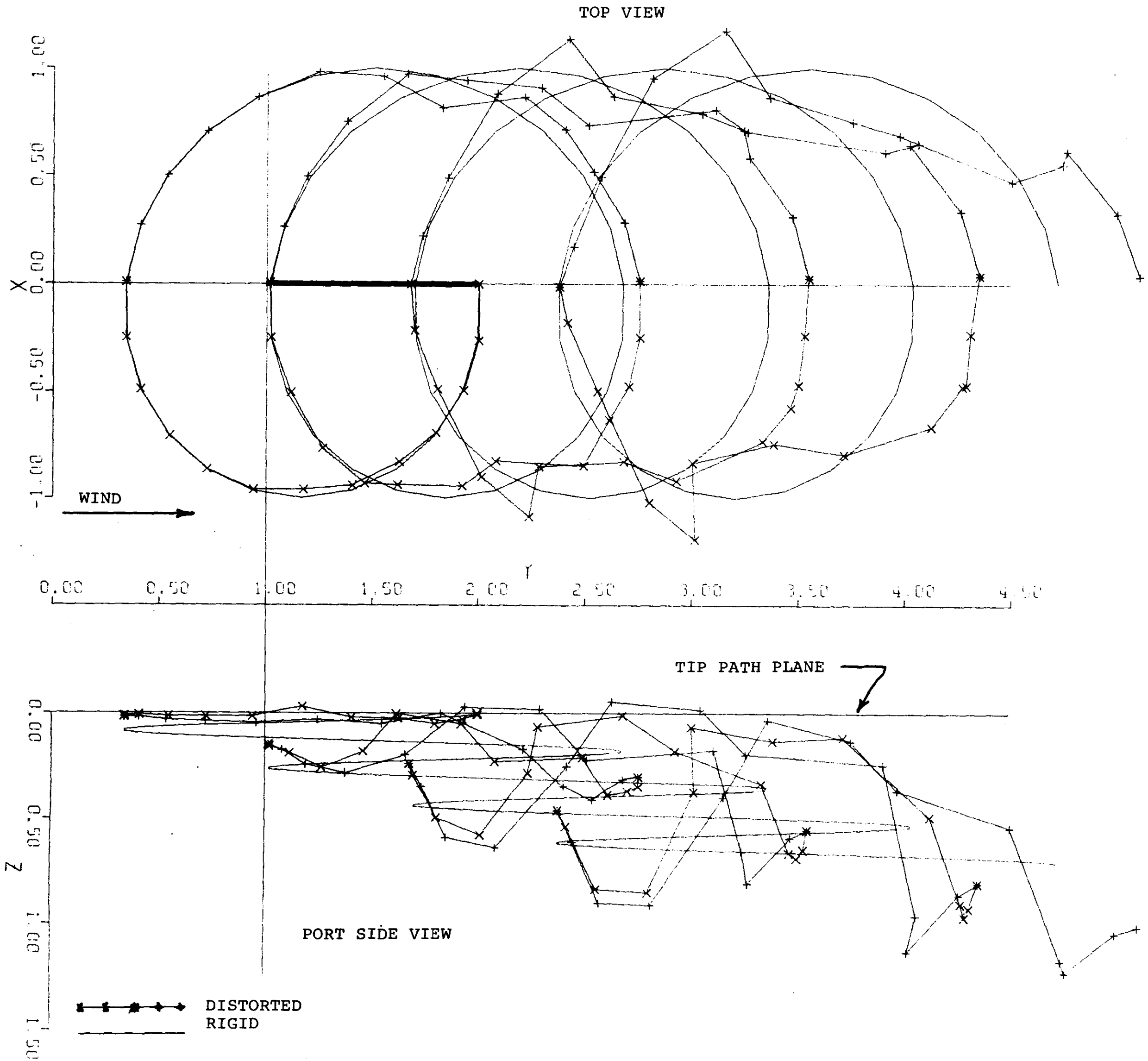

FIG. 37b RIGID VS. DISTORTED TIP VORTEX GEOMETRIES ψ = 0° BLADE, 6-BLADE ROTOR, μ = 0.108

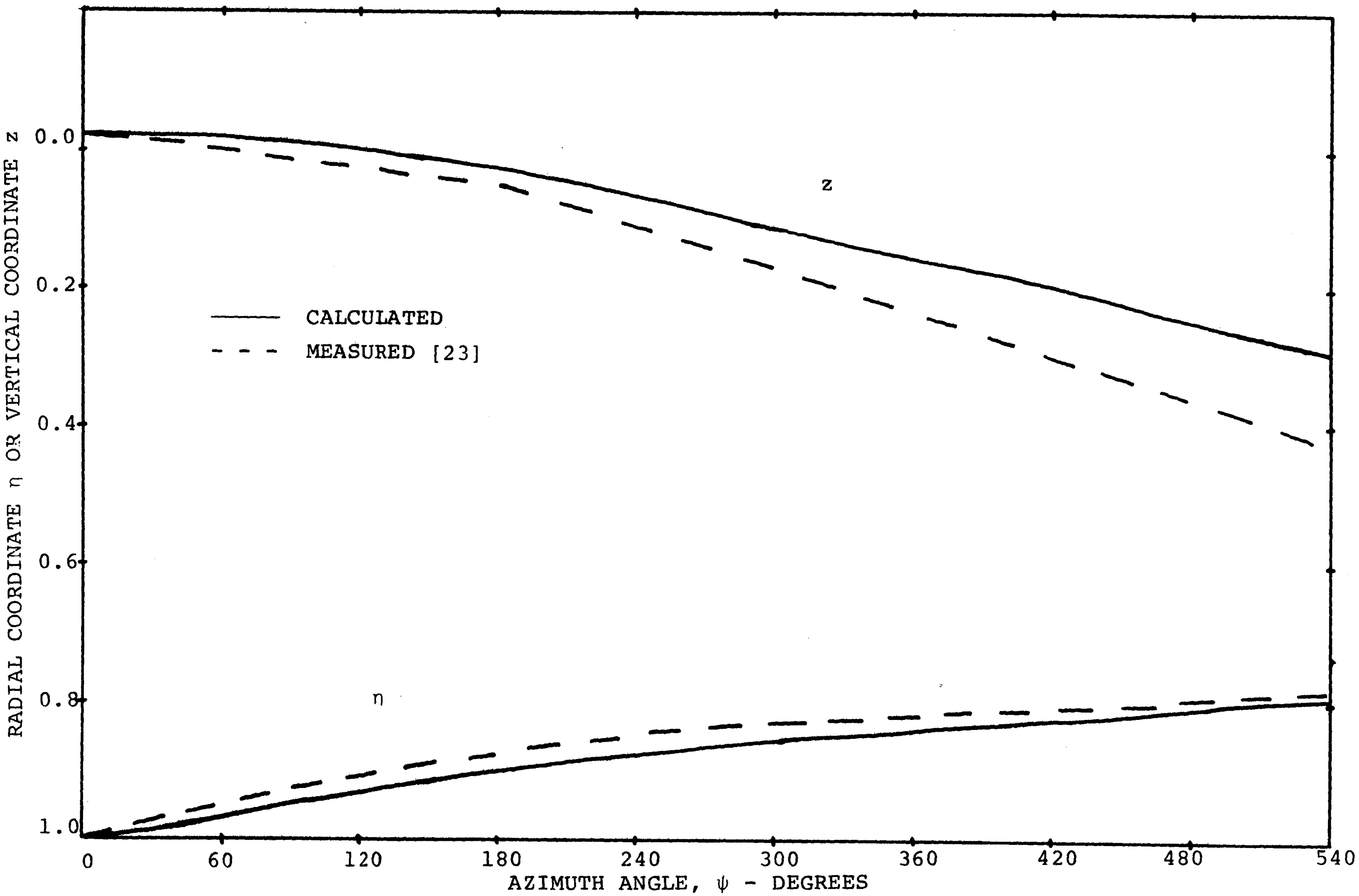

FIG. 38 HOVERING TIP VORTEX GEOMETRY

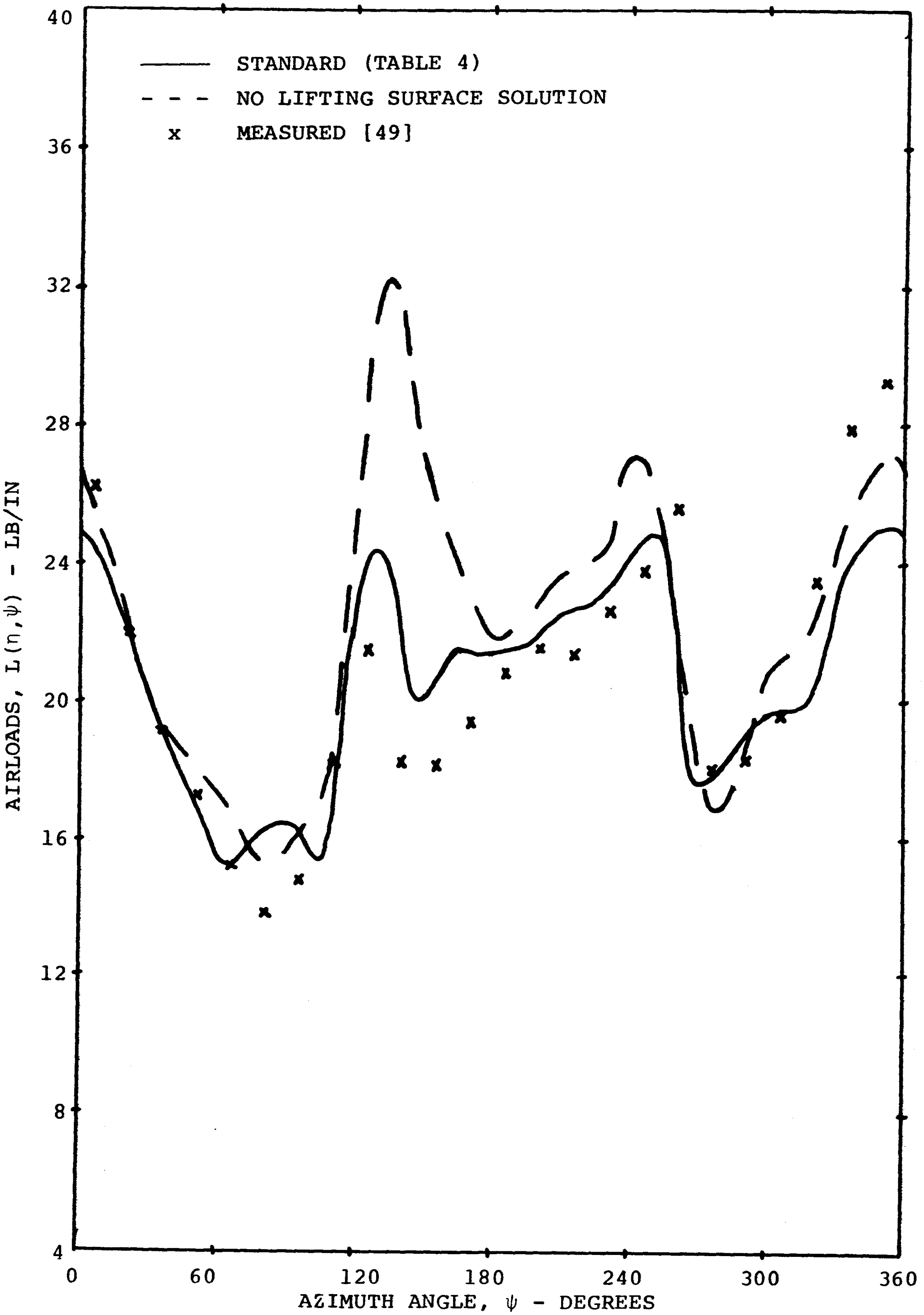

FIG. 39 RIGID WAKE AIRLOADS, EFFECT OF LIFTING SURFACE SOLUTION, 85% RADIUS, 4-BLADE ROTOR, μ = 0.18

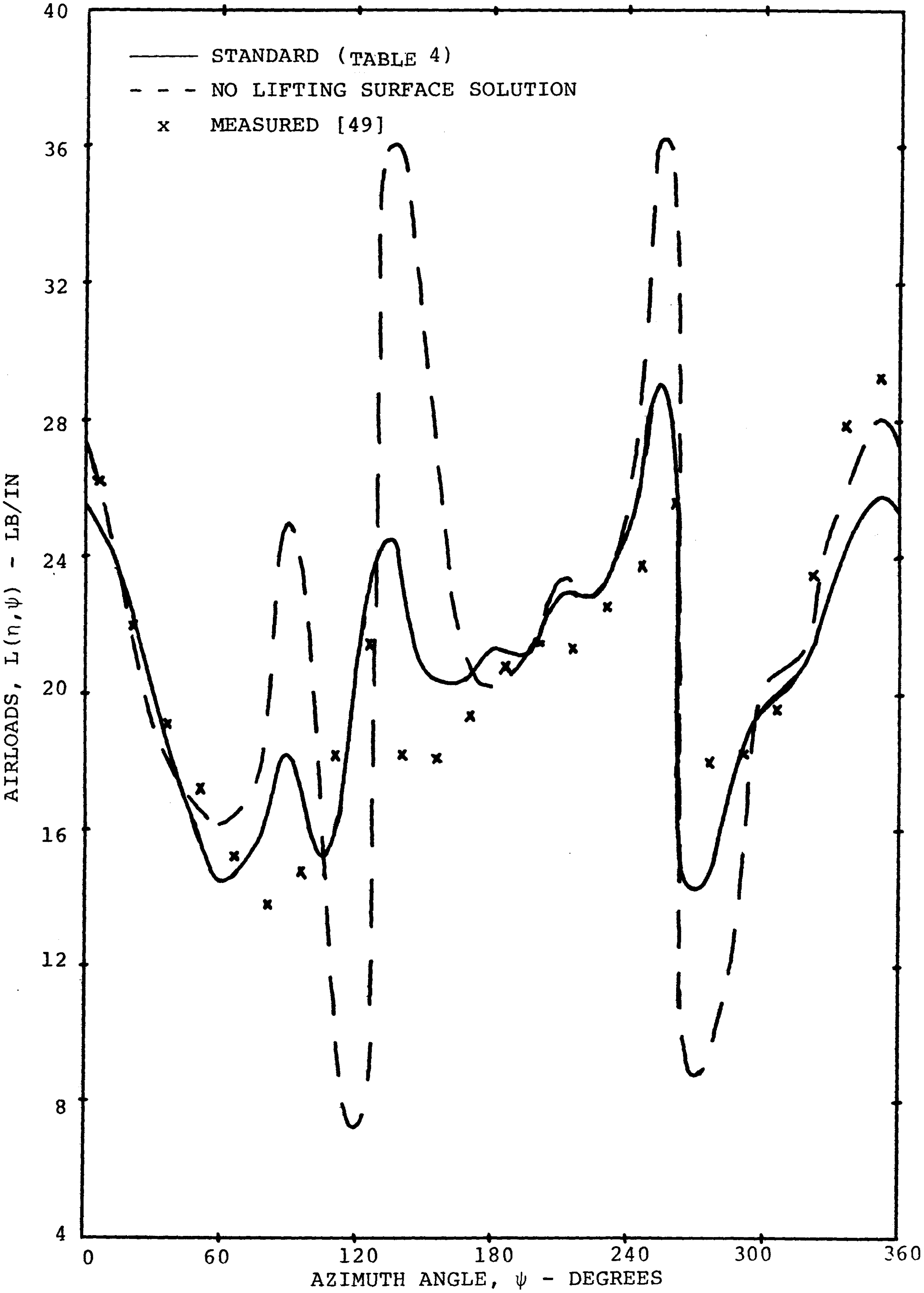

FIG. 40 DISTORTED WAKE AIRLOADS, EFFECT OF LIFTING SURFACE SOLUTION, 85% RADIUS, 4-BLADE ROTOR, $\mu = 0.18$

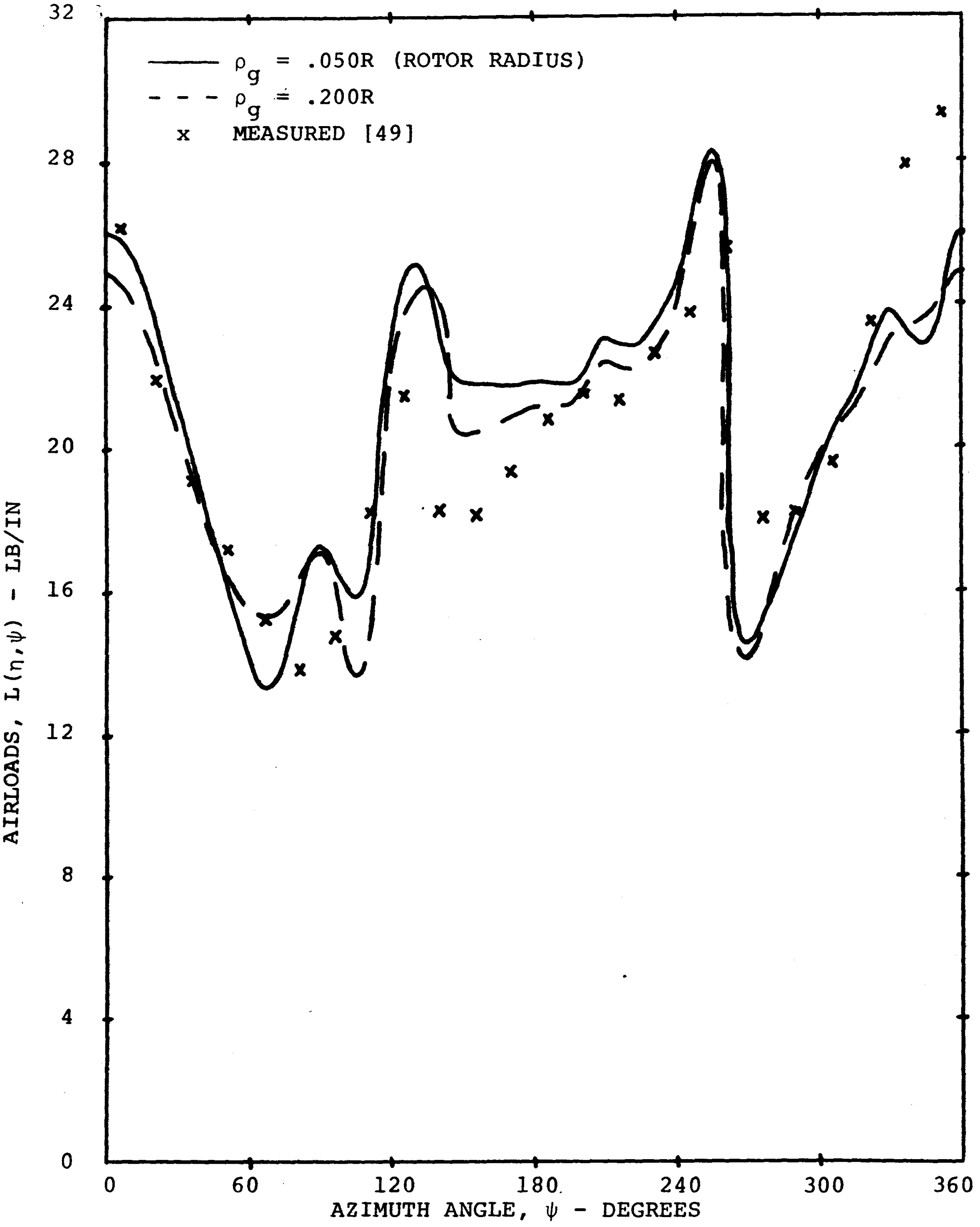

FIG. 41 DISTORTED WAKE AIRLOADS, BURST VORTEX CORE RADIUS ρ_g = .050R VS ρ_g = .200R, 85% RADIUS, 4-BLADE ROTOR, μ = 0.18

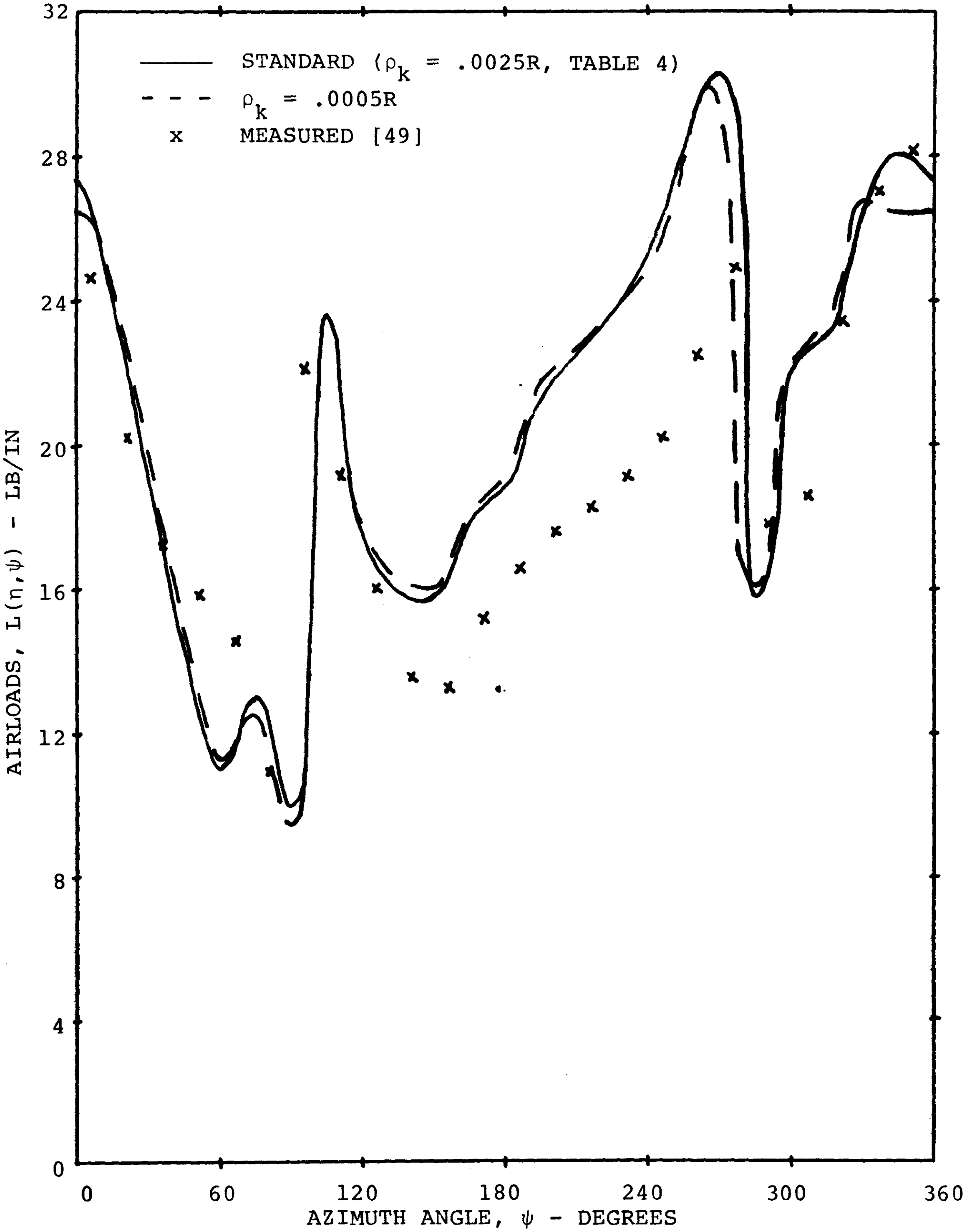

FIG. 42 DISTORTED WAKE AIRLOADS, EFFECT OF INITIAL VORTEX CORE RADIUS ρ_k = .0025R VS. ρ_k = .0005R, 95% RADIUS 4-BLADE ROTOR, μ = 0.18

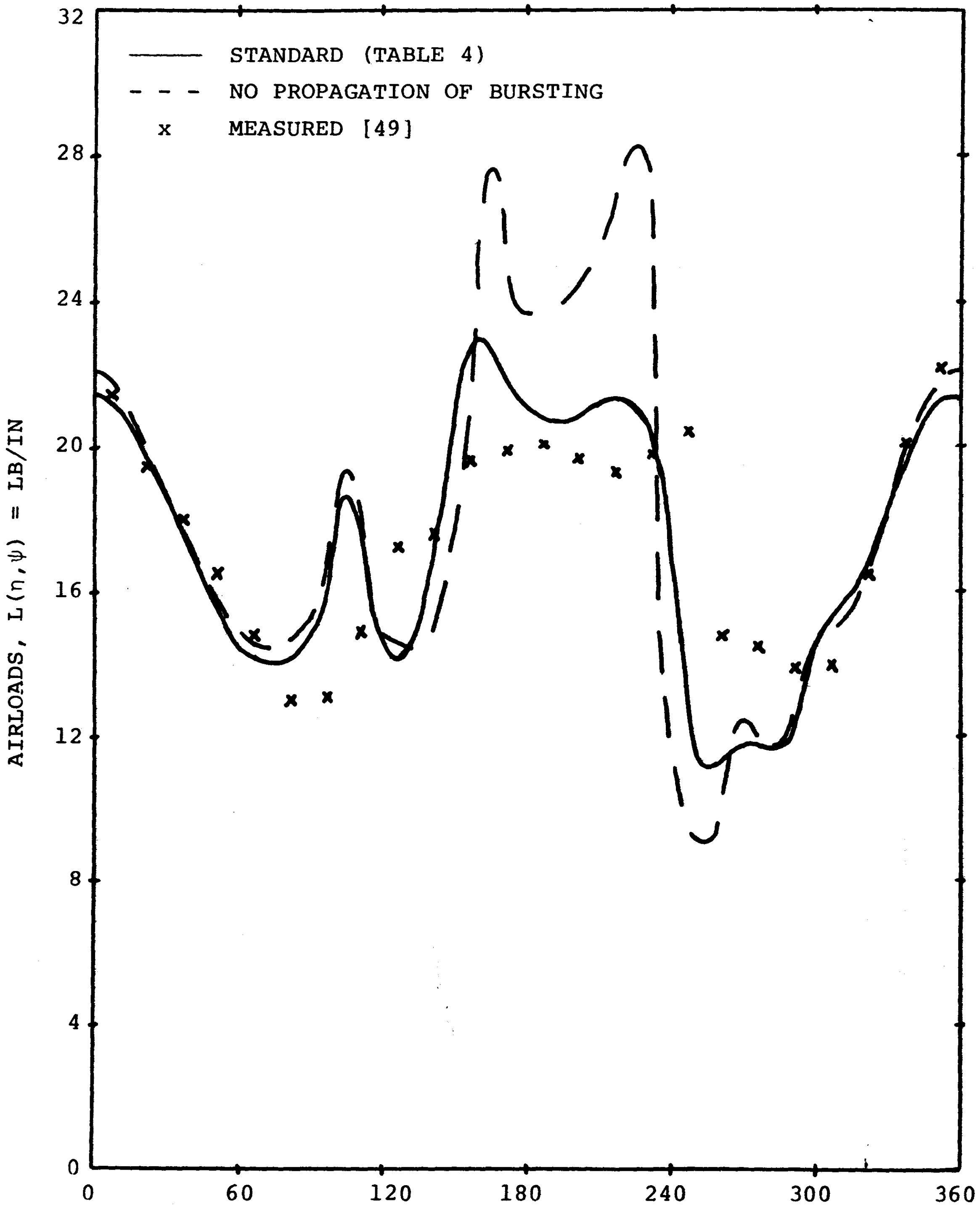

FIG. 43 DISTORTED WAKE AIRLOADS, EFFECT OF PROPAGATION OF TIP VORTEX CORE BURSTING, 75% RADIUS, 4-BLADE ROTOR, $\mu = 0.18$

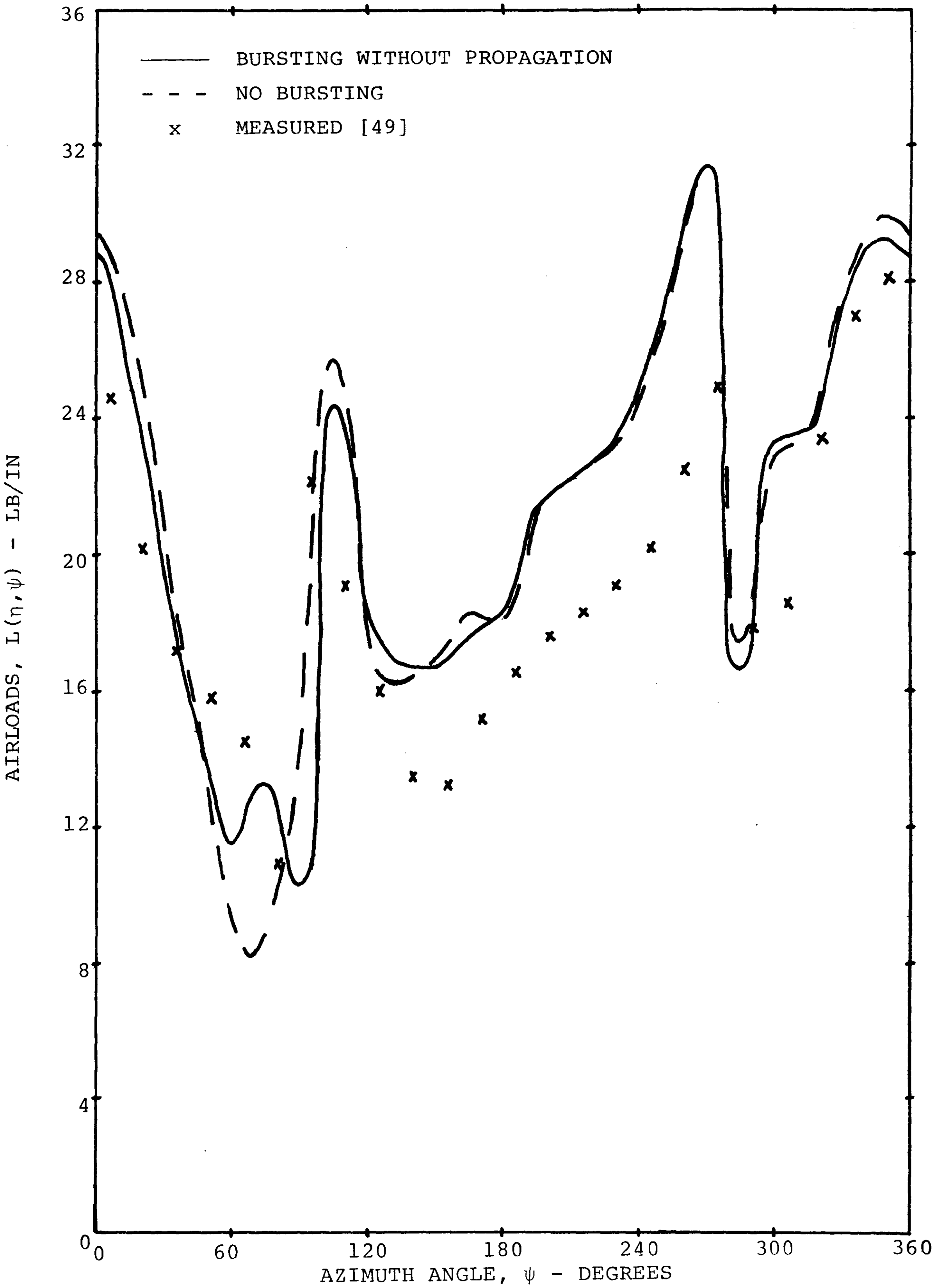

FIG. 44a DISTORTED WAKE AIRLOADS, TIP VORTEX CORE BURSTING WITHOUT PROPAGATION VS. NO BURSTING, 95% RADIUS, 4-BLADE ROTOR, μ = 0.18

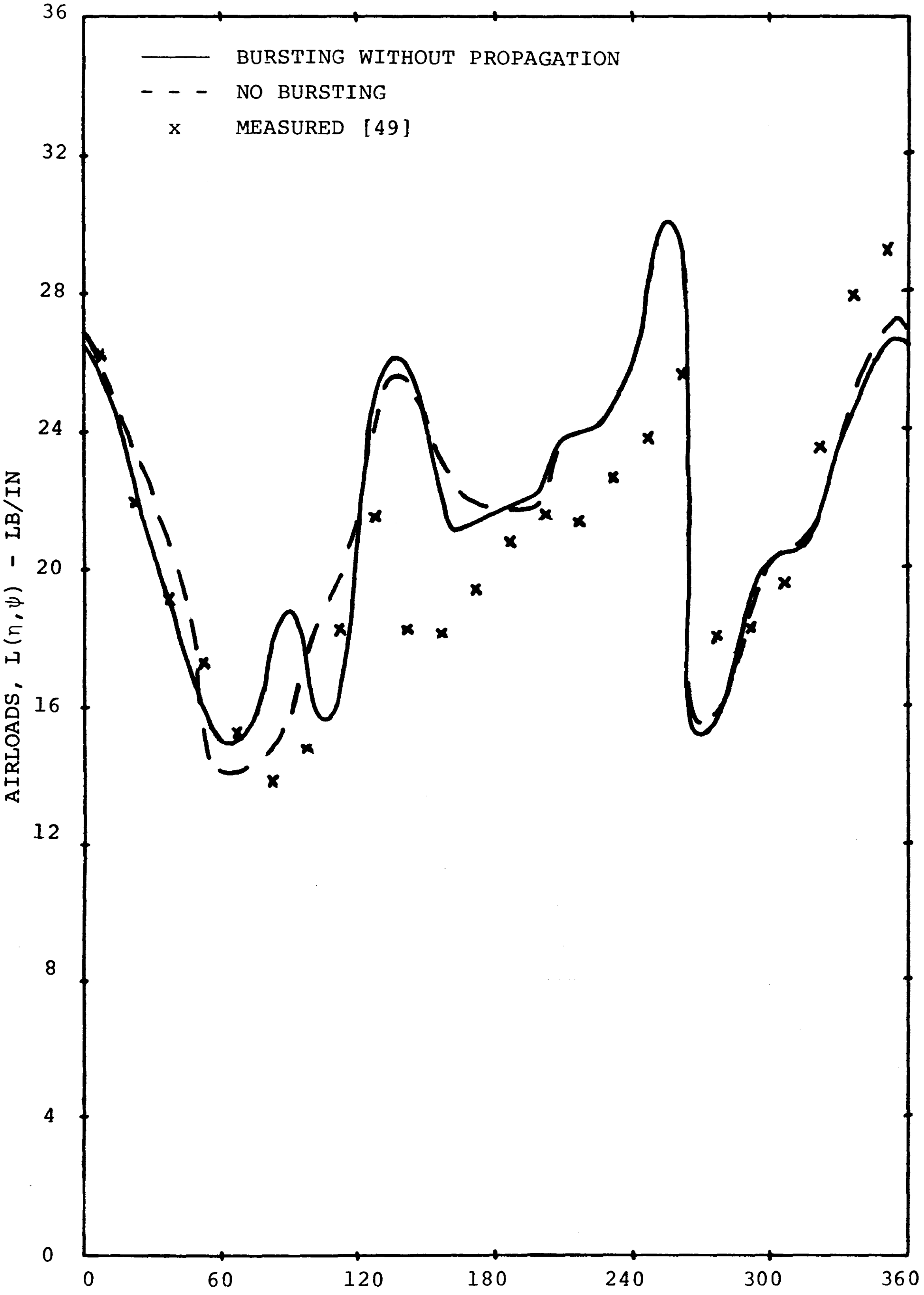

FIG. 44b DISTORTED WAKE AIRLOADS, TIP VORTEX-CORE BURSTING WITHOUT PROPAGATION VS. NO BURSTING, 85% RADIUS, 4-BLADE ROTOR, μ = 0.18

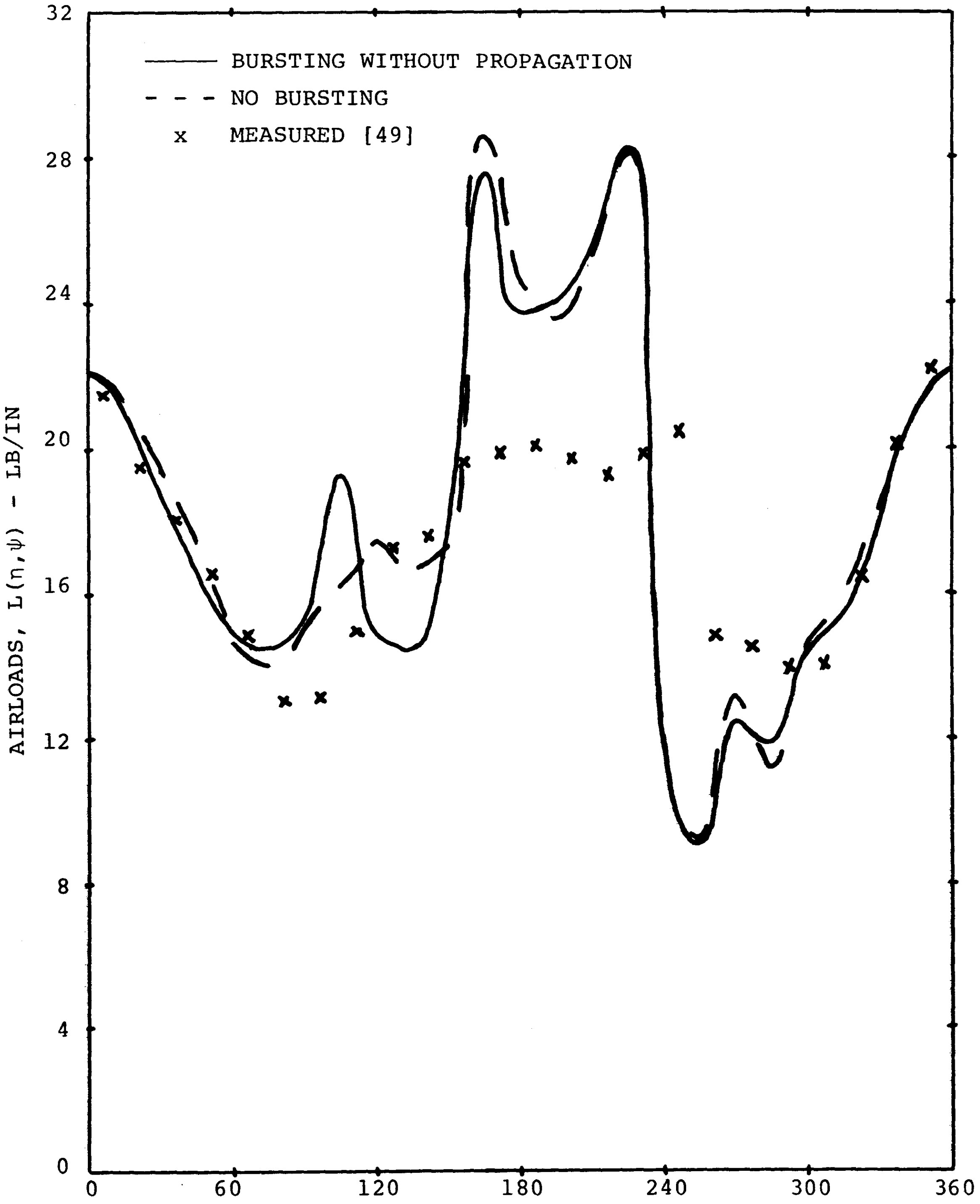

FIG. 44c DISTORTED WAKE AIRLOADS, TIP VORTEX CORE BURSTING VS. NO BURSTING, 75% RADIUS, 4-BLADE ROTOR, $\mu = 0.18$

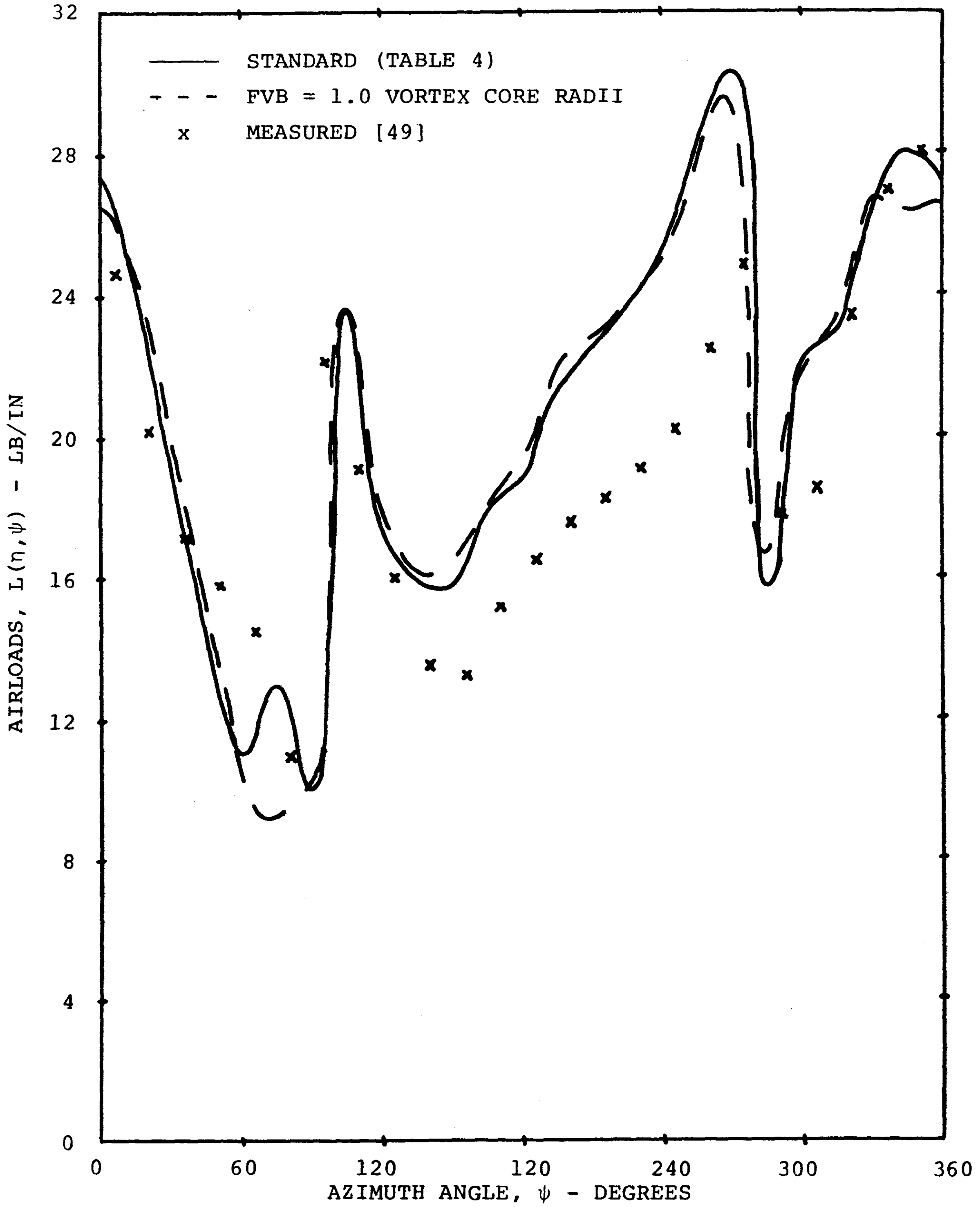

FIG. 45 DISTORTED WAKE AIRLOADS, EFFECT OF BLADE-VORTEX VERTICAL SEPARATION CRITERION FVB = 1.0 VORTEX CORE RADII, 95% RADIUS, 4-BLADE ROTOR, μ = 0.18

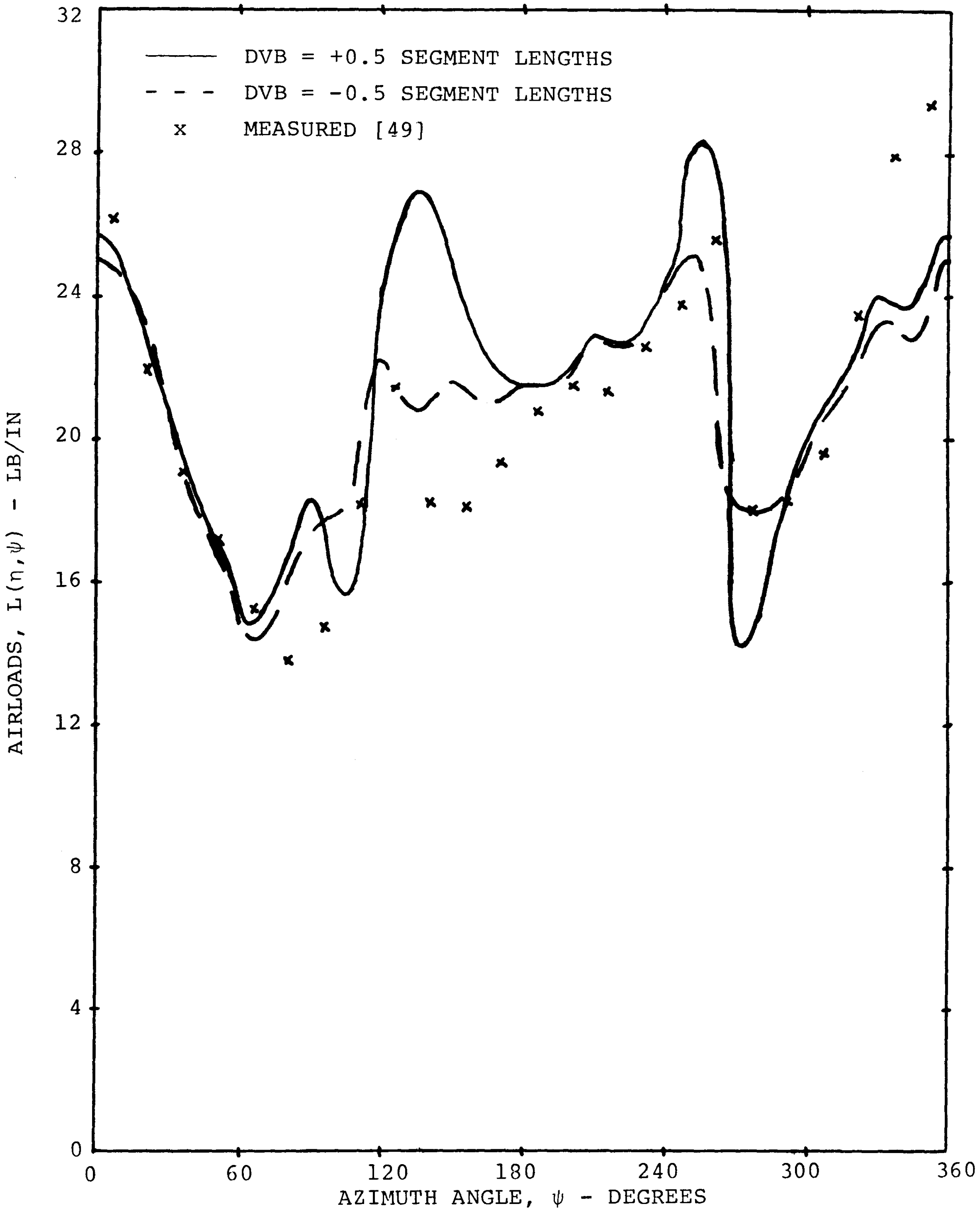

FIG. 46 DISTORTED WAKE AIRLOADS, EFFECT OF HORIZONTAL VORTEX BURSTING CRITERION DVB, 85% RADIUS, 4-BLADE ROTOR, $\mu = 0.18$

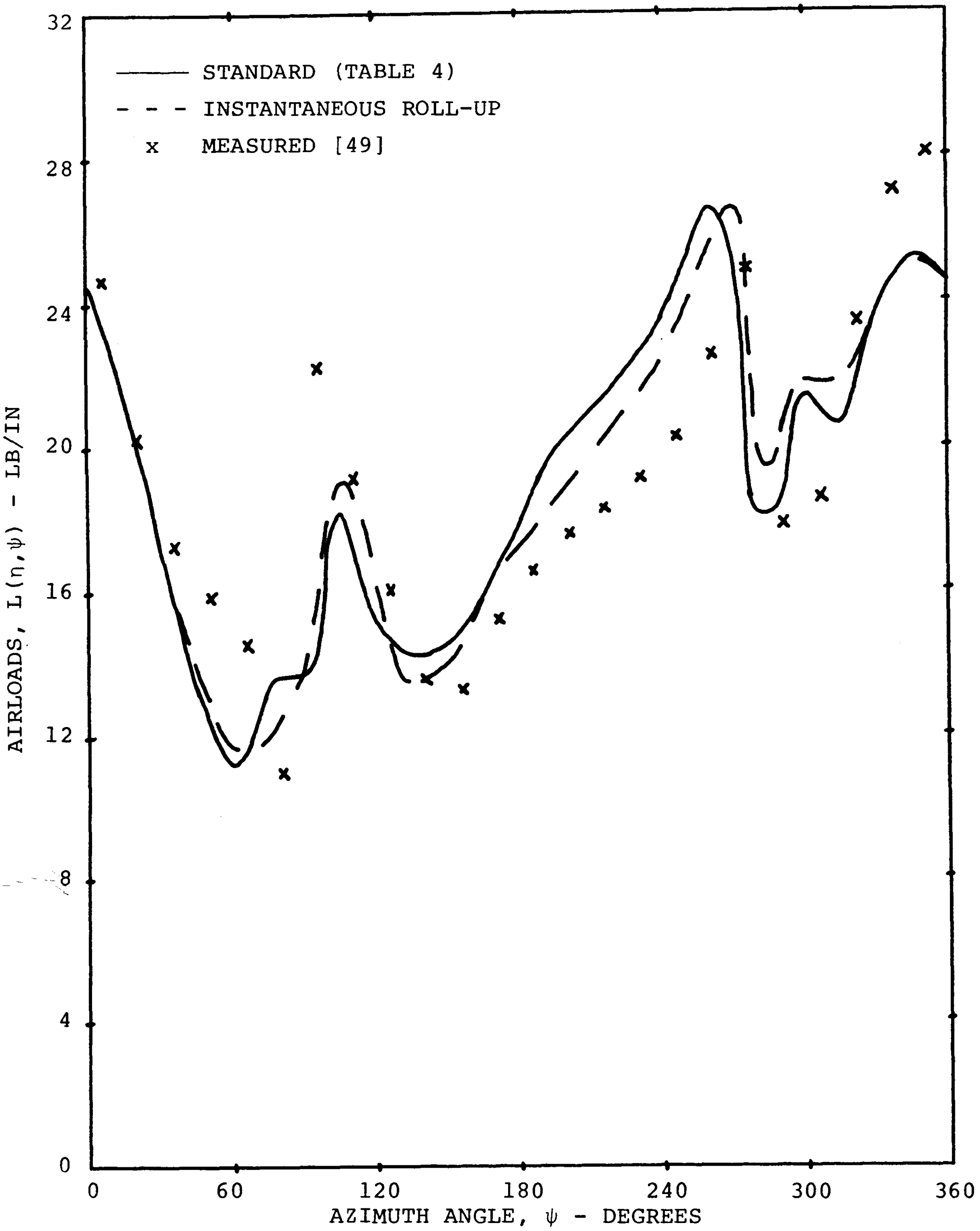

FIG. 47a RIGID WAKE AIRLOADS, EFFECT OF INSTANTANEOUS TIP VORTEX ROLL-UP, 95% RADIUS, 4-BLADE ROTOR, $\mu = 0.18$

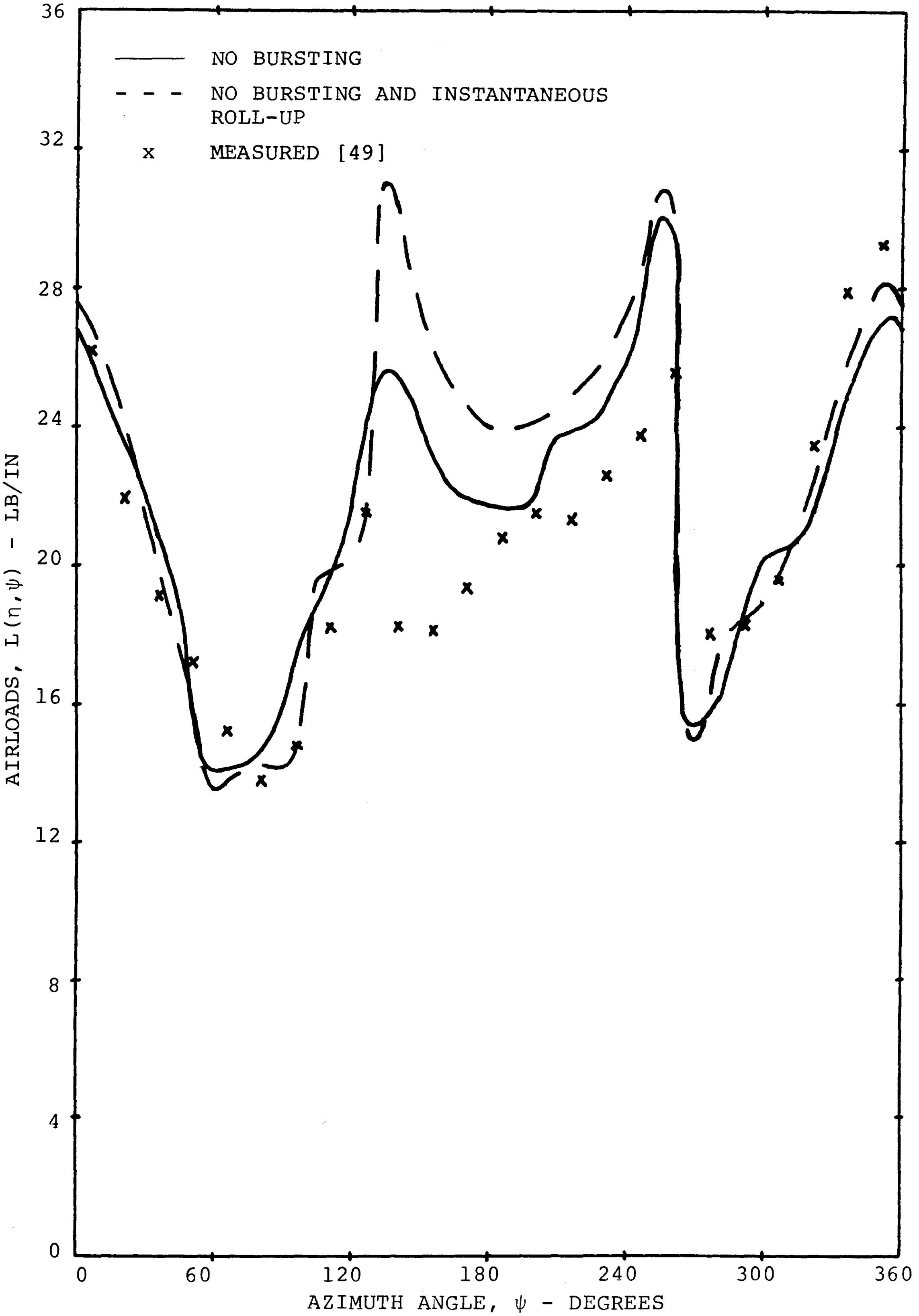

FIG. 47b DISTORTED WAKE AIRLOADS, EFFECT OF INSTANTANEOUS TIP VORTEX ROLL UP ON THE NO-BURSTING CASE, 85% RADIUS 4-BLADE ROTOR, $\mu = 0.18$

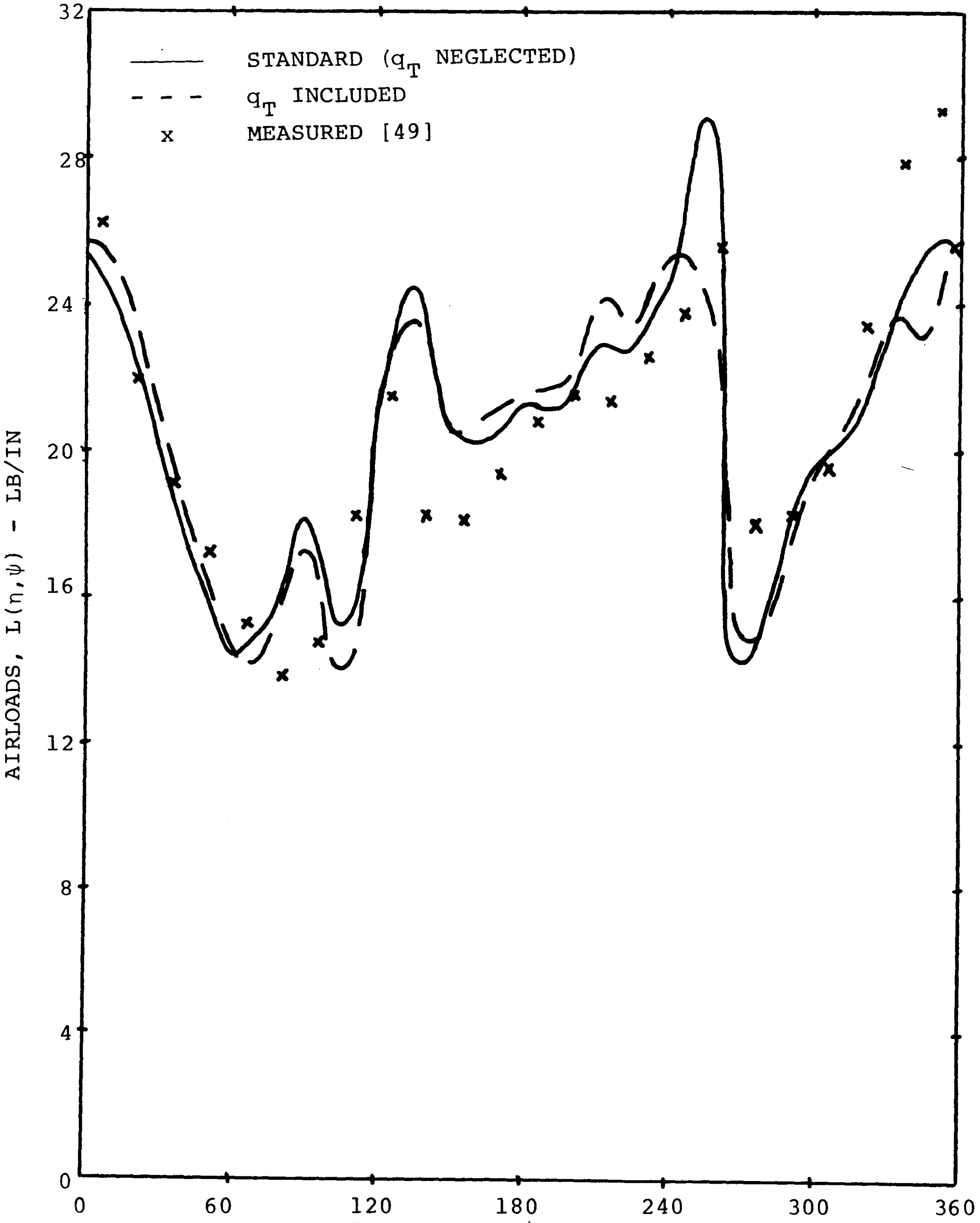

FIG. 48 DISTORTED WAKE AIRLOADS, EFFECT OF TANGENTIALLY INDUCED VELOCITY (q_T), 85% RADIUS, 4-BLADE ROTOR, $\mu = 0.18$

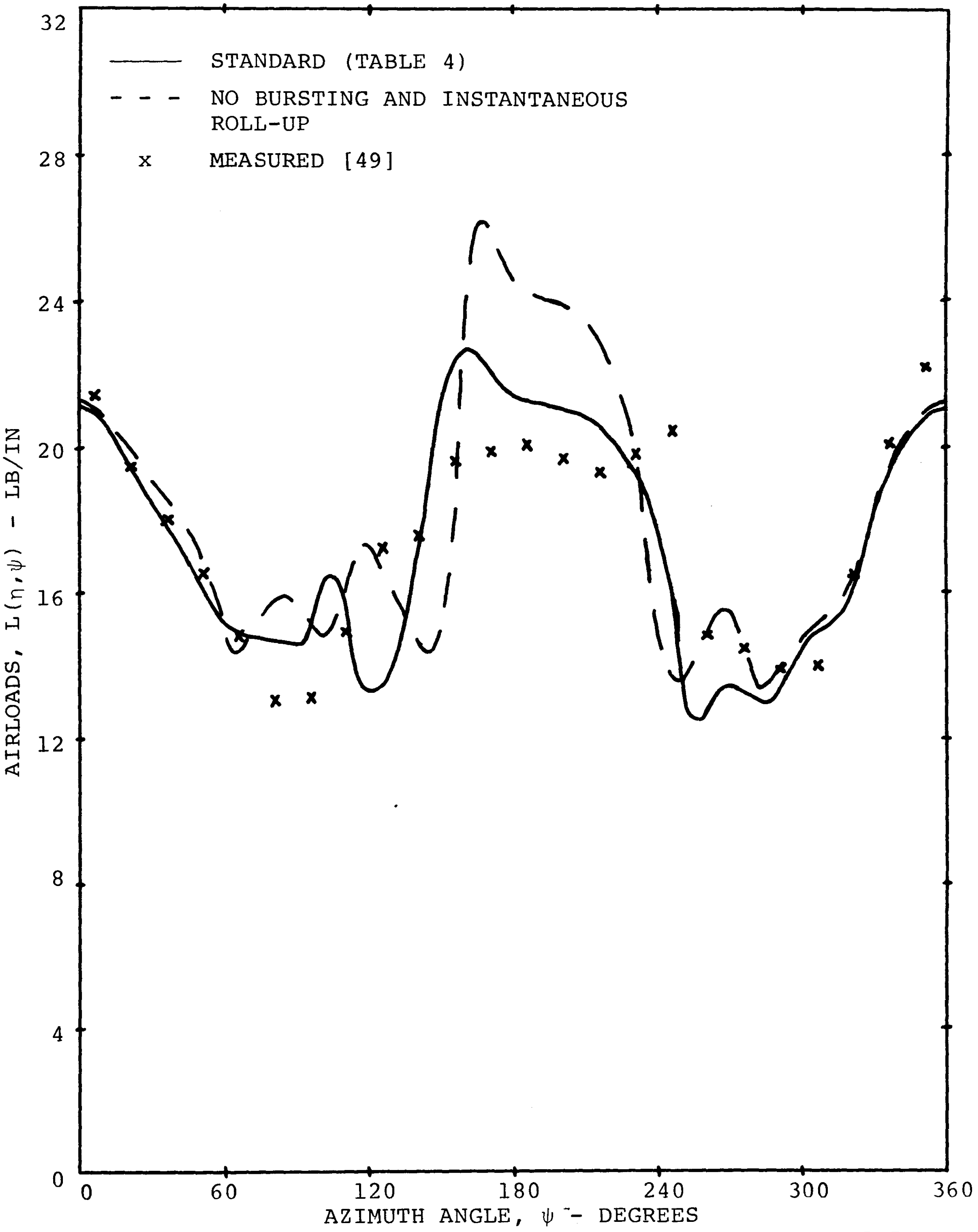

FIG. 49 RIGID WAKE AIRLOADS, STANDARD CASE VS. NO BURSTING AND INSTANTANEOUS ROLL-UP, 75% RADIUS, 4-BLADE ROTOR, $\mu = 0.18$

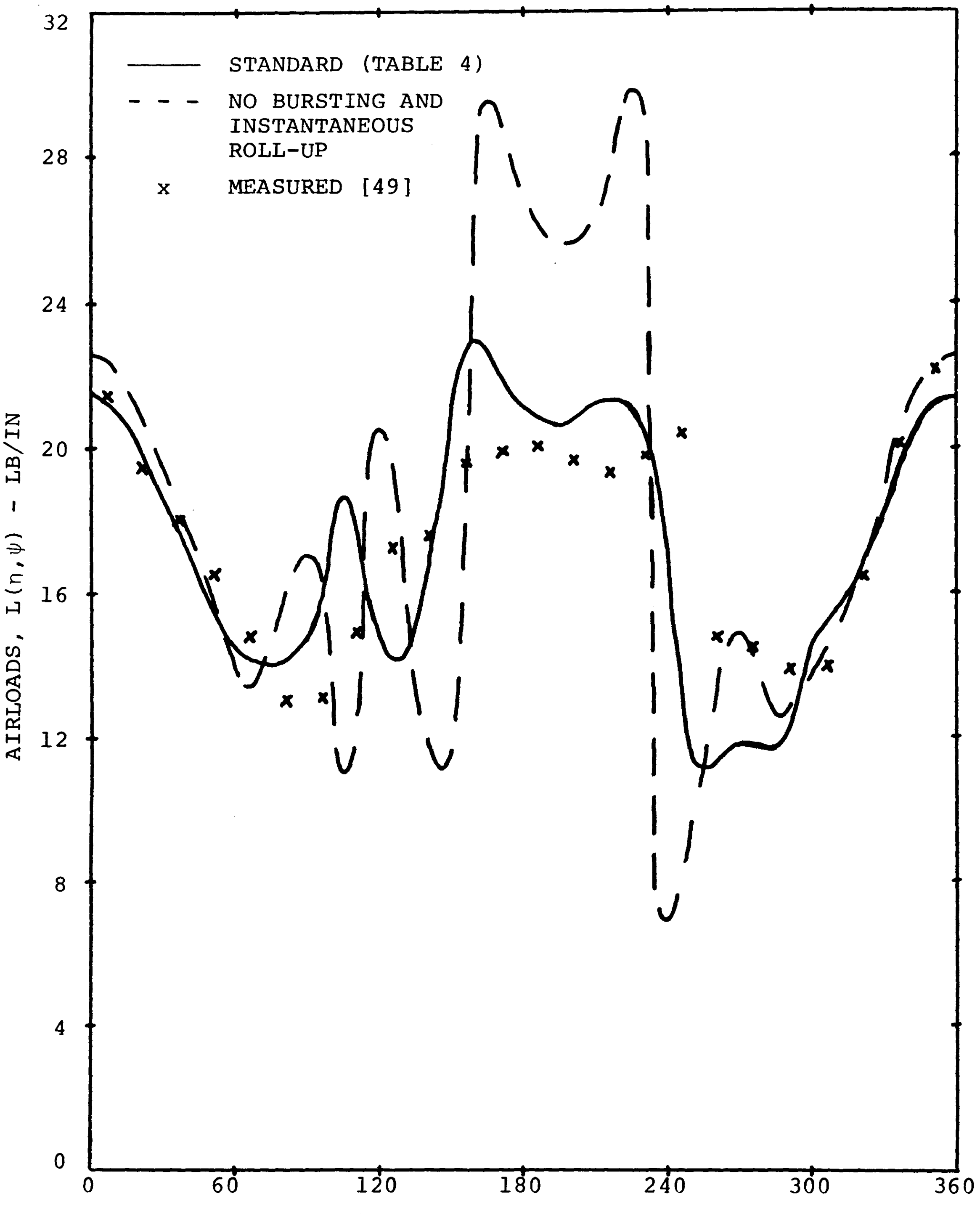

FIG. 50 DISTORTED WAKE AIRLOADS, STANDARD CASE VS. NO BURSTING AND INSTANTANEOUS ROLL-UP, 75% RADIUS, 4-BLADE ROTOR, $\mu = 0.18$

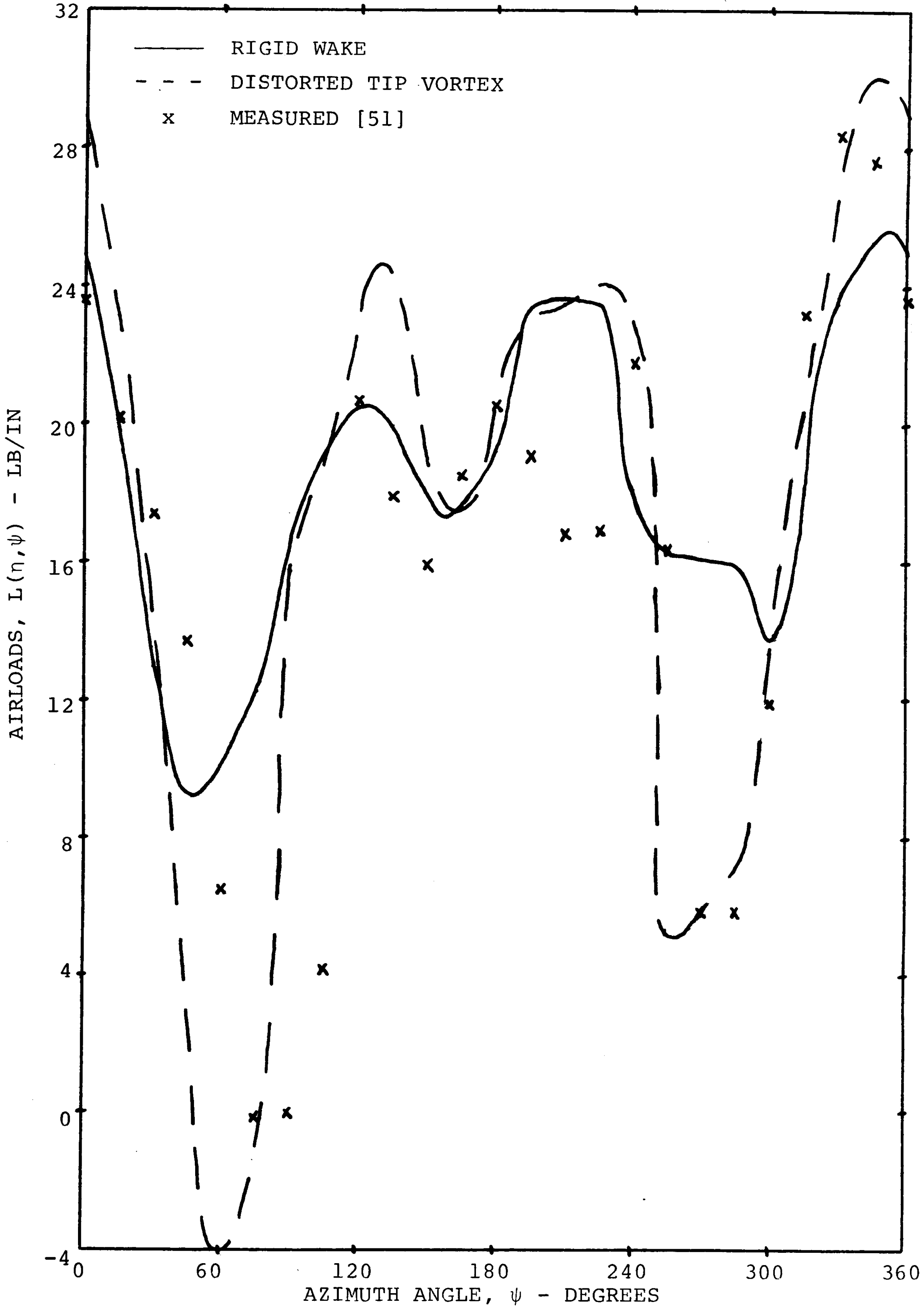

FIG. 51 RIGID VS. DISTORTED WAKE AIRLOADS, 85% RADIUS, 6-BLADE ROTOR, $\mu = 0.108$

APPENDIX A

VELOCITY INDUCED BY A STRAIGHT VORTEX LINE SEGMENT

Consider a straight vortex line segment. Label the ends of the vortex line segment points P_a and P_b. The nondimensional circulation is allowed to vary linearly from γ_a at point P_a to γ_b at point P_b. The induced velocity is computed at point P_η. Point P_η is located by vectors $\vec{a}$ and $\vec{b}$ from point P_η to points P_a and P_b (Fig. A.1). The TPP referenced x, y, z coordinate system is used with the origin at point P_η.

The induced velocity is computed by using the Biot Savart Relation [34]:

$$dq_1 = \frac{\Gamma \sin\theta \; ds_1}{4\pi r_1^2} \tag{A.1}$$

where dq_1 is the induced velocity at point P_η and is perpendicular to the r_1, ds_1 plane, ds_1 is an element of the vortex line, Γ is the circulation of ds_1, r_1 is the distance from point P_η to ds_1, and θ is the angle between r_1 and ds_1 (Fig. A.1). Now define the following nondimensional versions of the variables:

$$dq_{VL} \equiv dq_1 / \Omega R \tag{A.2}$$

$$r \equiv r_1 / R \tag{A.3}$$

$$ds \equiv ds_1 / R \tag{A.4}$$

$$\gamma \equiv \Gamma / 2\pi b \Omega R \tag{A.5}$$

Further, define h as the perpendicular distance (nondimensional by R) from point P_η to the vortex line segment and ϕ as the angle between h and r (Fig. A.1). Now substituting in Eq. A.1:

$$dq_{VL} = \gamma\left(\frac{b}{2R}\right) \frac{\sin\theta \, ds}{r^2} \tag{A.6}$$

From Fig. A.1:

$$\sin\theta = \cos\phi = \frac{h}{r} \tag{A.7}$$

$$ds = d(h\tan\phi) = \left(\frac{h}{\cos^2\phi}\right) d\phi \tag{A.8}$$

Now Eq. A.6 can be rewritten in terms of h and ϕ:

$$dq_{VL} = \gamma\left(\frac{b}{2R}\right)\left(\frac{\cos\phi}{h}\right) d\phi \tag{A.9}$$

Since γ is a linear variation from γ_a at point P_a to γ_b at point P_b, it can be written as (see Fig. A.1):

$$\gamma = \gamma_a + (\gamma_b - \gamma_a)\left(\frac{h\cot\alpha + h\tan\phi}{h\cot\alpha + h\cot\beta}\right) \tag{A.10}$$

Substituting into Eq. A.9 and integrating from $\phi = \alpha - \pi/2$ to $\phi = \pi/2 - \beta$:

$$q_{VL} = \gamma_a\left(\frac{b}{2R}\right)\int_{\alpha-\frac{\pi}{2}}^{\frac{\pi}{2}-\beta}\left\{\frac{\cos\phi}{h} + \left(\frac{\gamma_b-\gamma_a}{\gamma_a}\right)\left[\frac{\cot\alpha\cos\phi + \sin\phi}{h(\cot\alpha + \cot\beta)}\right]\right\} d\phi \tag{A.11}$$

$$q_{VL} = \gamma_a\left(\frac{b}{2R}\right)\left\{\left(\frac{\cos\alpha + \cos\beta}{h}\right) + \left(\frac{\gamma_b-\gamma_a}{\gamma_a}\right)\left[\frac{\sin\alpha - \sin\beta + \cot\alpha(\cos\alpha + \cos\beta)}{h(\cot\alpha + \cot\beta)}\right]\right\} \tag{A.12}$$

For the special case of constant circulation ($\gamma_a = \gamma_b$) this reduces to:

$$q_{VL} = \gamma_a\left(\frac{b}{2R}\right)\left(\frac{\cos\alpha + \cos\beta}{h}\right) \tag{A.13}$$

To find q_{VL} in terms of $\vec{a}$ and $\vec{b}$, define $\vec{c}$ (Fig. A.1).

$$\vec{c} \equiv \vec{b} - \vec{a} \tag{A.14}$$

Now using the dot and cross product relations and Fig. A.1:

$$-\vec{a}\cdot\vec{c} = (-\vec{a})\cdot\vec{c} = |\vec{a}||\vec{c}|\cos\alpha \tag{A.15}$$

$$\vec{b}\cdot\vec{c} = (-\vec{c})\cdot(-\vec{b}) = |\vec{b}||\vec{c}|\cos\beta \tag{A.16}$$

$$|\vec{a}\times\vec{c}| = |\vec{a}||\vec{c}|\sin\alpha \tag{A.17}$$

$$|\vec{b}\times\vec{c}| = |\vec{b}||\vec{c}|\sin\beta \tag{A.18}$$

$$h = |\vec{a}|\sin\alpha = \frac{|\vec{a}\times\vec{c}|}{|\vec{c}|} \tag{A.19}$$

Using the definition of $\vec{c}$ and recalling that a vector crossed with itself is zero:

$$|\vec{a}\times\vec{c}| = |\vec{a}\times(\vec{b}-\vec{a})| = |\vec{a}\times\vec{b}| \tag{A.20}$$

$$|\vec{b}\times\vec{c}| = |\vec{b}\times(\vec{b}-\vec{a})| = |\vec{a}\times\vec{b}| \tag{A.21}$$

Recall that the induced velocity is perpendicular to the $r_1 ds_1$ plane, which is the same as the $\vec{a}$, $\vec{b}$ plane. Define positive circulation such that $\vec{q}_{VL}$ points into the paper in Fig. A.1. Now $\vec{q}_{VL}$ can be written as:

$$\vec{q}_{VL} = q_{VL}\frac{\vec{a}\times\vec{b}}{|\vec{a}\times\vec{b}|} \tag{A.22}$$

Substituting into Eq. A.12:

$$\vec{q}_{VL} = \gamma_a\left(\frac{b}{2R}\right)\left\{\left[\frac{(|\vec{a}|+|\vec{b}|)(|\vec{a}||\vec{b}|-\vec{a}\cdot\vec{b})}{|\vec{a}||\vec{b}||\vec{a}\times\vec{b}|^2}\right]\left[1+\left(\frac{\gamma_b-\gamma_a}{\gamma_a}\right)\left(\frac{|\vec{a}|^2-\vec{a}\cdot\vec{b}}{|\vec{a}|^2+|\vec{b}|^2-2\vec{a}\cdot\vec{b}}\right)\right]\right.$$
$$\left.+\left(\frac{\gamma_b-\gamma_a}{\gamma_a}\right)\left[\frac{|\vec{b}|-|\vec{a}|}{|\vec{a}||\vec{b}|(|\vec{a}|^2+|\vec{b}|^2-2\vec{a}\cdot\vec{b})}\right]\right\}(\vec{a}\times\vec{b}) \tag{A.23}$$

For the special case of constant circulation ($\gamma_a = \gamma_b$), this reduces to:

$$\vec{q}_{VL} = \gamma_a\left(\frac{b}{2R}\right)\left[\frac{(|\vec{a}|+|\vec{b}|)(|\vec{a}||\vec{b}|-\vec{a}\cdot\vec{b})}{|\vec{a}||\vec{b}||\vec{a}\times\vec{b}|^2}\right](\vec{a}\times\vec{b}) \tag{A.24}$$

APPENDIX B

VELOCITY INDUCED BY A PLANE RECTANGULAR VORTEX SHEET

B.1 Computation of the Induced Velocity in Vortex-Sheet Based Coordinates

Consider a plane, rectangular vortex sheet whose circulation is aligned parallel to one pair of edges, hereafter called the front and back edges. The circulation is allowed to vary linearly from γ_a to γ_b parallel to the circulation vector but remains constant perpendicular to the circulation vector. The induced velocity is computed at point P_η. Point P_η is located by vectors $\vec{a}$ and $\vec{b}$ from point P_η to the ends of the forward edge of the vortex sheet, points P_a and P_b, and by the perpendicular distance $z_\perp$ from point P_η to the vortex sheet (Fig. B.1).

A right-handed, rectangular coordinate system $(\tilde{x}, \tilde{y}, \tilde{z})$ with its origin at point P_η is chosen. The $\tilde{z}$ axis is perpendicular to the vortex sheet while $\tilde{x}$ and $\tilde{y}$ are parallel to it. The $\tilde{y}$ axis is parallel to the front and back edges, while the $\tilde{x}$ axis points forward perpendicular to the front edge (Fig. B.1). The length of the vortex sheet in the $\tilde{x}$ direction is $\Delta\tilde{x}$ with the front edge at $\tilde{x} = x_1$ and the back edge at $\tilde{x} = x_2$. The width of the vortex sheet in the $\tilde{y}$ direction is $|\vec{c}| = |\vec{b} - \vec{a}|$ (Fig. B.1).

To compute the induced velocity contribution of the vortex sheet at point P_η $(\vec{q}_{VS})$, the sheet is visualized as an infinite number of parallel vortex lines each of infinitesimal circulation. The geometry of the vortex line at $\tilde{x} = x_1$ relative to point P_η is specified by $\vec{a}$ and $\vec{b}$. To compute the induced velocity contribution of this vortex line, it is helpful to define the parameters c_a, c_b, h_1, $\vec{n}_1$, α_1, and β_1 with the aid of Fig. B.1. The perpendicular distance from point P_η to the vortex line $(\vec{c})$ or its extension is h_1. The distances from the intersection of h_1 with $\vec{c}$ or its extension to points P_a and P_b are c_a and c_b, respectively. The angles between $\vec{a}$ and $\vec{c}$ and between $\vec{b}$ and $\vec{c}$ are α_1 and β_1, respectively.

The circulation of the vortex sheet segment (nonD by $2\pi b\Omega R$) varies linearly from γ_a at the P_a edge of the sheet to γ_b at the P_b edge. Spreading this over $\Delta\tilde{x}$, the circulation of each infinitesimal vortex line varies from $[(\gamma_a/\Delta\tilde{x})\ d\tilde{x}]$ at the P_a end to $[(\gamma_b/\Delta\tilde{x}\ d\tilde{x})]$ at the P_b end. Using the expression for the induced velocity of a straight vortex line segment from Eq. A.14, the induced velocity contribution of the vortex line at $\tilde{x} = x_1$ (nondimensional by tip speed) is:

$$dq_1 = \left(\frac{b}{2R}\right)\left(\frac{\gamma_a}{\Delta\tilde{x}}\right) d\tilde{x}\ \vec{n} \left\{ \left(\frac{\cos\alpha_1 + \cos\beta_1}{h_1}\right) + \left(\frac{\gamma_b - \gamma_a}{\gamma_a}\right)\left[\frac{\sin\alpha_1 - \sin\beta_1 + \cot\alpha_1(\cos\alpha_1 + \cos\beta_1)}{h_1(\cot\alpha_1 + \cot\beta_1)}\right]\right\} \tag{B.1}$$

For the special case of a constant circulation strength vortex line ($\gamma_a = \gamma_b$) only the first term remains.

Now consider the induced velocity contribution of a general vortex line at $\tilde{x}$. Since the vortex sheet is rectangular, c_a and c_b remain constant for all $\tilde{x}$. The general values of h, $\vec{n}$, α, and β are, however, functions of $\tilde{x}$ and are denoted by dropping the subcript 1 from the values at $\tilde{x} = x_1$. The induced velocity contribution of the vortex sheet is obtained by integration of the vortex line contributions in the positive $\tilde{x}$ direction from $\tilde{x} = x_2$ to $\tilde{x} = x_1$. This gives:

$$\vec{q}_{vs} = \left(\frac{b}{2R}\right)\left(\frac{\gamma_a}{\Delta\tilde{x}}\right)\left\{ \int_{x_2}^{x_1} \left(\frac{\cos\alpha + \cos\beta}{h}\right)\vec{n}\ d\tilde{x} + \left(\frac{\gamma_b - \gamma_a}{\gamma_a}\right)\int_{x_2}^{x_1}\left[\frac{\sin\alpha - \sin\beta + \cot\alpha(\cos\alpha + \cos\beta)}{h(\cot\alpha + \cot\beta)}\right]\vec{n}\ d\tilde{x}\right\} \tag{B.2}$$

To integrate Eq. 8.2, h, $\vec{n}$, α, and β must be expressed in terms of $\tilde{x}$. From Fig. B.1:

$$h = \sqrt{\tilde{x}^2 + z_\perp^2} \tag{B.3}$$

$$\sin\alpha = \frac{h}{\sqrt{h^2 + c_a^2}} = \frac{\sqrt{\tilde{x}^2 + z_\perp^2}}{\sqrt{\tilde{x}^2 + z_\perp^2 + c_a^2}} \tag{B.4}$$

$$\sin\beta = \frac{h}{\sqrt{h^2 + c_b^2}} = \frac{\sqrt{\tilde{x}^2 + z_\perp^2}}{\sqrt{\tilde{x}^2 + z_\perp^2 + c_b^2}} \tag{B.5}$$

$$\cos\alpha = \frac{c_a}{\sqrt{\tilde{x}^2 + z_\perp^2 + c_a^2}} \tag{B.6}$$

$$\cos\beta = \frac{c_b}{\sqrt{\tilde{x}^2 + z_\perp^2 + c_a^2}} \tag{B.7}$$

Observing that $\vec{n}$ is perpendicular to h and in the plane formed by h and $z_\perp$:

$$\vec{n} = \frac{-z_\perp}{\sqrt{\tilde{x}^2 + z_\perp^2}}\,\vec{i}_{\tilde{x}} + \frac{\tilde{x}}{\sqrt{\tilde{x}^2 + z_\perp^2}}\,\vec{i}_{\tilde{z}} \tag{B.8}$$

where $\vec{i}_{\tilde{x}}$ and $\vec{i}_{\tilde{z}}$ are unit vectors in the $\tilde{x}$ and $\tilde{z}$ directions, respectively. Equation B.2 can now be rewritten in terms of $\tilde{x}$ as:

$$\begin{aligned}\vec{q}_{vs} = \left(\frac{b}{2R}\right)\left(\frac{\gamma_a}{\Delta\tilde{x}}\right)\Bigg\{ &\int_{x_2}^{x_1}\left(\frac{c_a}{\sqrt{c_a^2 + \tilde{x}^2 + z_\perp^2}} + \frac{c_b}{\sqrt{c_b^2 + \tilde{x}^2 + z_\perp^2}}\right)\left(\frac{-z_\perp\vec{i}_{\tilde{x}} + \tilde{x}\,\vec{i}_{\tilde{z}}}{\tilde{x}^2 + \tilde{z}^2}\right)d\tilde{x} \\ &+ \left(\frac{\gamma_b - \gamma_a}{\gamma_a}\right)\int_{x_2}^{x_1}\Bigg[\frac{1}{\sqrt{c_a^2 + \tilde{x}^2 + z_\perp^2}} - \frac{1}{\sqrt{c_b^2 + \tilde{x}^2 + z_\perp^2}} \\ &+ \left(\frac{c_a}{\tilde{x}^2 + z_\perp^2}\right)\left(\frac{c_a}{\sqrt{c_a^2 + \tilde{x}^2 + z_\perp^2}} + \frac{c_b}{\sqrt{c_b^2 + \tilde{x}^2 + z_\perp^2}}\right)\Bigg]\left(\frac{-z_\perp\vec{i}_{\tilde{x}} + \tilde{x}\,\vec{i}_{\tilde{z}}}{c_a + c_b}\right)d\tilde{x}\Bigg\}\end{aligned} \tag{B.9}$$

Equation B.9 has four fundamentally different integrals in it:

$$I_1 = \int_{x_2}^{x_1} \left[\left(\frac{C_a}{\sqrt{C_a^2 + \tilde{x}^2 + z_\perp}} + \frac{C_b}{\sqrt{C_b^2 + \tilde{x}^2 + z_\perp^2}} \right) \left(\frac{-z_\perp \, d\tilde{x}}{\tilde{x}^2 + z_\perp^2} \right) \right] \tag{B.10}$$

$$I_2 = \int_{x_2}^{x_1} \left[\left(\frac{C_a}{\sqrt{C_a^2 + \tilde{x}^2 + z_\perp^2}} + \frac{C_b}{\sqrt{C_b^2 + \tilde{x}^2 + z_\perp^2}} \right) \left(\frac{\tilde{x} \, d\tilde{x}}{\tilde{x}^2 + z_\perp^2} \right) \right] \tag{B.11}$$

$$I_3 = \int_{x_2}^{x_1} \left[\left(\frac{1}{\sqrt{C_a^2 + \tilde{x}^2 + z_\perp^2}} - \frac{1}{\sqrt{C_b^2 + \tilde{x}^2 + z_\perp^2}} \right) \left(-z_\perp \, d\tilde{x} \right) \right] \tag{B.12}$$

$$I_4 = \int_{x_2}^{x_1} \left[\left(\frac{1}{\sqrt{C_a^2 + \tilde{x}^2 + z_\perp^2}} - \frac{1}{\sqrt{C_b^2 + \tilde{x}^2 + z_\perp^2}} \right) \left(\tilde{x} \, d\tilde{x} \right) \right] \tag{B.13}$$

The integrals can be looked up in a table or integrated directly. To simplify writing the results define:

$$S_{1a} = \sqrt{C_a^2 + x_1^2 + z_\perp^2} \tag{B.14}$$

$$S_{1b} = \sqrt{C_b^2 + x_1^2 + z_\perp^2} \tag{B.15}$$

$$S_{2a} = \sqrt{C_a^2 + x_2^2 + z_\perp^2} \tag{B.16}$$

$$S_{2b} = \sqrt{C_b^2 + x_2^2 + z_\perp^2} \tag{B.17}$$

The integrals are evaluated at x_1 and x_2 and algebraic manipulation is used to minimize the number of natural log functions which saves computer time.

$$I_1 = \tan^{-1}\left(\frac{C_a X_2}{z_\perp S_{2a}}\right) - \tan^{-1}\left(\frac{C_a X_1}{z_\perp S_{1a}}\right) + \tan^{-1}\left(\frac{C_b X_2}{z_\perp S_{2b}}\right) - \tan^{-1}\left(\frac{C_b X_1}{z_\perp S_{1b}}\right) \tag{B.18}$$

$$I_2 = \frac{1}{2}\left[\ln\left(\frac{S_{1a} - C_a}{S_{1a} + C_a}\right) - \ln\left(\frac{S_{2a} - C_a}{S_{2a} + C_a}\right) + \ln\left(\frac{S_{1b} - C_b}{S_{1b} + C_b}\right) - \ln\left(\frac{S_{2b} - C_b}{S_{2b} + C_b}\right)\right]$$

$$= \frac{1}{2}\ln\left[\frac{(S_{1a} - C_a)(S_{2a} + C_a)(S_{1b} - C_b)(S_{2b} + C_b)}{(S_{1a} + C_a)(S_{2a} - C_a)(S_{1b} + C_b)(S_{2b} - C_b)}\right] \tag{B.19}$$

$$I_3 = \ln(S_{2a} + X_2) - \ln(S_{1a} + X_1) - \ln(S_{2b} + X_2) + \ln(S_{1b} + X_1)$$

$$= \ln\left[\frac{(S_{2a} + X_2)(S_{1b} + X_1)}{(S_{1a} + X_1)(S_{2b} + X_2)}\right] \tag{B.20}$$

$$I_4 = S_{1a} - S_{2a} - S_{1b} + S_{2b} \tag{B.21}$$

Finally, the induced velocity contribution of the vortex sheet at point P_η (nondimensional by tip speed) can be written as:

$$\vec{q}_{vs} = \left(\frac{b}{2R}\right)\left(\frac{\gamma_a}{\Delta\tilde{x}}\right)\left\{ I_1\,\vec{i}_{\tilde{x}} + I_2\,\vec{i}_{\tilde{z}} + \left(\frac{\gamma_b - \gamma_a}{\gamma_a}\right)\right.$$

$$\left.\left[\left(\frac{C_a}{C_a + C_b}\right)\left(I_1\,\vec{i}_{\tilde{x}} + I_2\,\vec{i}_{\tilde{z}}\right) + \left(\frac{1}{C_a + C_b}\right)\left(I_3\,\vec{i}_{\tilde{x}} + I_4\,\vec{i}_{\tilde{z}}\right)\right]\right\} \tag{B.22}$$

For the special case of a constant circulation vortex sheet ($\gamma_a = \gamma_b$) this simplifies to:

$$\vec{q}_{vs} = \left(\frac{b}{2R}\right)\left(\frac{\gamma_a}{\Delta\tilde{x}}\right)\left(I_1\,\vec{i}_x + I_2\,\vec{i}_z\right) \tag{B.23}$$

Notice that I_1 (the $\tan^{-1}$ integral) has no singularities, but I_2 (the ℓn integral) is singular near the edges of the vortex sheet. I_1 can be said to represent the fundamental nature of the vortex sheet and I_2 to represent the edges. To explore the edge singularity, consider the cases: $x_1 = \varepsilon$, $\Delta\tilde{x} = \Delta x$, $c_a = c_b = c$, $z_\perp = 0$, and $\varepsilon << \Delta x$, c. From Eq. B.19:

$$I_2 = \frac{1}{2}\ln\left[\frac{(S_1-c)^2(S_2+c)^2}{(S_1+c)^2(S_2-c)^2}\right] = \ln\left[\frac{(S_1-c)(S_2+c)}{(S_1+c)(S_2-c)}\right] \tag{B.24}$$

Expanding S_1 and S_2 in ε:

$$S_1 = \sqrt{c^2+\varepsilon^2} = c + \frac{\varepsilon^2}{2c} + HOT \tag{B.25}$$

$$\begin{aligned} S_2 &= \sqrt{c^2+\Delta x^2 - 2\varepsilon\Delta x + \varepsilon^2} \\ &= \sqrt{c^2+\Delta x^2} + \frac{\varepsilon(\varepsilon-2\Delta x)}{2\sqrt{c^2+\Delta x^2}} + HOT \end{aligned} \tag{B.26}$$

Dropping higher-order terms (HOT) and plugging in to Eq. B.24:

$$I_2 \simeq \ln\left[\frac{\left(\sqrt{c^2+\Delta x^2} + \frac{\varepsilon(\varepsilon-2\Delta x)}{2\sqrt{c^2+\Delta x^2}} + c\right)\left(c + \frac{\varepsilon^2}{2c} - c\right)}{\left(\sqrt{c^2+\Delta x^2} + \frac{\varepsilon(\varepsilon-2\Delta x)}{2\sqrt{c^2+\Delta x^2}} - c\right)\left(c + \frac{\varepsilon^2}{2c} + c\right)}\right] \tag{B.27}$$

Now, for ε very, very small:

$$I_2 \simeq \ln\left[\left(\frac{\sqrt{c^2+\Delta x^2}+c}{\sqrt{c^2+\Delta x^2}-c}\right)\left(\frac{\varepsilon}{2c}\right)^2\right] \doteq 2\ln\left(\frac{\varepsilon}{2c}\right) \tag{B.28}$$

A similar exercise with $\varepsilon << \Delta x << c$ results in $I_2 \sim \ell n(\varepsilon/\Delta x)$. Thus there is a logarithmic singularity when x_1 or x_2 becomes small compared with the dimensions of the vortex sheet. The way to avoid this is to stay away from the edges of the sheet, since in reality there are no such step changes in circulation.

B.2 Transformation from Vortex Sheet Based Coordinates to TPP Coordinates

To be able to add the induced velocity contributions of the many vortex sheets of the wake model, a common coordinate system must be used. Since vortex sheet orientation varies greatly, the $\tilde{x}$, $\tilde{y}$, $\tilde{z}$ system is not satisfactory. Since $\vec{a}$ and $\vec{b}$ are already available in the x, y, z (TPP referenced) coordinate system and all vortex line segment computations are done in this coordinate system it is chosen. To review, it is a right-handed, rectangular coordinate system where the z axis points downward perpendicular to the TPP, the y axis points downstream and parallel to the TPP, and the x axis points to starboard parallel to the TPP.

It is also necessary to express the vortex sheet parameters c_a, c_b, $z_\perp$, x_1, x_2, and $\Delta\tilde{x}$ in terms of parameters similar to those used in vortex line computations. The vectors $\vec{a}$ and $\vec{b}$ from point P_η to points P_a and P_b on the forward edge of the vortex sheet are analogous to the $\vec{a}$ and $\vec{b}$ from point P_η to points P_a and P_b at the ends of a vortex line segment. In addition, the length of the vortex sheet $\Delta\tilde{x}$ and z_m, the z component of a vector from point P_η to the midpoint of the $\tilde{x} = x_2$ edge of the vortex sheet are used to completely specify the vortex sheet geometry.

Let $\vec{i}$, $\vec{j}$, $\vec{k}$ be unit vectors along the x, y, z axes, respectively. Now $\vec{a}$ and $\vec{b}$ can be written in the x, y, x coordinate system as:

$$\vec{a} = x_a \vec{i} + y_a \vec{j} + z_a \vec{k} \tag{B.29}$$

$$\vec{b} = x_b \vec{i} + y_b \vec{j} + z_b \vec{k} \tag{B.30}$$

Recall the definition of $\vec{c}$ (Fig. B.2):

$$\vec{c} \equiv \vec{b} - \vec{a} \tag{B.31}$$

Define:

$$\Delta x \equiv x_b - x_a \tag{B.32}$$

$$\Delta y \equiv y_b - y_a \tag{B.33}$$

$$\Delta z \equiv z_b - z_a \tag{B.34}$$

Therefore:

$$\vec{c} = \Delta x\, \vec{i} + \Delta y\, \vec{j} + \Delta z\, \vec{k} \tag{B.35}$$

$$|\vec{c}|^2 = \Delta x^2 + \Delta y^2 + \Delta z^2 \tag{B.36}$$

From Fig. B.2:

$$c_a = |\vec{a}| \sin \alpha \tag{B.37}$$

Now using the dot product relation:

$$\vec{c} \cdot (-\vec{a}) = -\vec{c} \cdot \vec{a} = |\vec{c}||\vec{a}| \sin \alpha \tag{B.38}$$

Therefore:

$$c_a = \frac{-\vec{c} \cdot \vec{a}}{|\vec{c}|} = \frac{-(x_a \Delta x + y_a \Delta y + z_a \Delta z)}{\sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}} \tag{B.39}$$

From Fig. B.2:

$$\vec{c}_b = |\vec{c}| - c_a \tag{B.40}$$

To find the relationship between the $\tilde{x}$, $\tilde{y}$, $\tilde{x}$ and the x, y, z coordinate systems, define the x, y, z components of the $\vec{i}_{\tilde{x}}$, $\vec{i}_{\tilde{y}}$, and $\vec{i}_{\tilde{z}}$ unit vectors:

$$\vec{i}_{\tilde{x}} = x_x \vec{i} + y_x \vec{j} + z_x \vec{k} \tag{B.41}$$

$$\vec{i}_{\tilde{y}} = x_y \vec{i} + y_y \vec{j} + z_y \vec{k} \tag{B.42}$$

$$\vec{i}_{\tilde{z}} = x_z \vec{i} + y_z \vec{j} + z_z \vec{k} \tag{B.43}$$

Now since $\vec{c}$ is parallel to the $\tilde{y}$ axis (Fig. B.2):

$$\vec{i}_{\tilde{y}} = \frac{\vec{c}}{|\vec{c}|} \tag{B.44}$$

$$x_y = \frac{\Delta x}{|\vec{c}|} \tag{B.45}$$

$$y_y = \frac{\Delta y}{|\vec{c}|} \tag{B.46}$$

$$z_y = \frac{\Delta z}{|\vec{c}|} \tag{B.47}$$

Consider a vector $\vec{s}$ running from the midpoint of the $\tilde{x} = x_2$ edge of the vortex sheet to the midpoint of the $\tilde{x} = x_1$ edge. The length of $\vec{s}$ is the length of the vortex sheet ($\Delta\tilde{x}$), (Fig. B.3). Since $\vec{s}$ is parallel to the $\tilde{x}$ axis, it is parallel to the unit vector $\vec{i}_{\tilde{x}}$. The z coordinate of the $\tilde{x} = x_2$ end of $\vec{s}$ is z_m from the definition of z_m. The z coordinate of the $\tilde{x} = x_1$ end of $\vec{s}$ is the average of the z coordinates of the two ends of that edge of the vortex sheet (x_a and x_b). Now the z component of $\vec{i}_{\tilde{x}}$ is the difference between the z coordinates of the two ends of $\vec{s}$ divided by the length of $\vec{s}$. Therefore:

$$z_x = \left[\left(\frac{z_a + z_b}{2}\right) - z_m\right] \Big/ \Delta\tilde{x} \tag{B.48}$$

To find the other two components of $\vec{i}_{\tilde{x}}$, recall that $\vec{i}_{\tilde{x}}$ is a unit vector. Therefore,

$$|\vec{i}_{\tilde{x}}| = x_x^2 + y_x^2 + z_x^2 = 1.0 \tag{B.49}$$

and that $\vec{i}_{\tilde{x}}$ is perpendicular to $\vec{i}_{\tilde{y}}$. Therefore:

$$\vec{i}_{\tilde{x}} \cdot \vec{i}_{\tilde{y}} = \frac{x_x \Delta x + y_x \Delta y + z_x \Delta z}{|\vec{c}|} = 0.0 \tag{B.50}$$

This gives two equations, B.49 and B.50 which can be solved for x_x and y_x with the result that:

$$x_x = \frac{-z_x \Delta x \Delta z + \Delta y \sqrt{(\Delta x^2 + \Delta y^2) - z_x^2 |\vec{c}|^2}}{\Delta x^2 + \Delta y^2} \tag{B.51}$$

$$y_x = \frac{-z_x \Delta y \Delta z - \Delta x \sqrt{(\Delta x^2 + \Delta y^2) - z_x^2 |\vec{c}|^2}}{\Delta x^2 + \Delta y^2} \tag{B.52}$$

To find $\vec{i}_{\tilde{z}}$, recall that for a right-handed coordinate system:

$$\vec{i}_{\tilde{z}} = \vec{i}_{\tilde{x}} \times \vec{i}_{\tilde{y}} \tag{B.53}$$

This relation implies the following components for $\vec{i}_{\tilde{z}}$:

$$x_z = \frac{y_x \Delta z - z_x \Delta y}{|\vec{c}|} \tag{B.54}$$

$$y_z = \frac{z_x \Delta x - x_x \Delta z}{|\vec{c}|} \tag{B.55}$$

$$z_z = \frac{x_x \Delta y - y_x \Delta z}{|\vec{c}|} \tag{B.56}$$

The parameters x_1, x_2, and $z_\perp$ remain to be derived in terms of x, y, z coordinates. To do this, consider a vector $\vec{h}$ which is the perpendicular dropped from point P_η to the $\tilde{x} = x_1$ edge of the vortex sheet (Fig. B.3). Note that the length of $\vec{h}$ is h_1, as was used in Section B.1 of this appendix.

Define the x, y, z components of $\vec{h}$ as:

$$\vec{h} = x_h \vec{i} + y_h \vec{j} + z_h \vec{k} \tag{B.57}$$

Now from Fig. B.2:

$$\vec{h} = \vec{a} + \vec{c}\left(\frac{c_a}{|\vec{c}|}\right) \tag{B.58}$$

This relation implies that:

$$x_h = x_a + \Delta x \left(\frac{c_a}{|\vec{c}|}\right) \tag{B.59}$$

$$y_h = y_a + \Delta y \left(\frac{c_a}{|\vec{c}|}\right) \tag{B.60}$$

$$z_h = z_a + \Delta z \left(\frac{c_a}{|\vec{c}|}\right) \tag{B.61}$$

From Fig. B.3:

$$x_1 = |\vec{h}| \cos\delta = h_1 \cos\delta \tag{B.62}$$

$$z_\perp = h_1 \cos\theta \tag{B.63}$$

From the dot product relation and Fig. B.3:

$$\vec{i}_{\tilde{x}} \cdot \vec{h} = |\vec{i}_{\tilde{x}}||\vec{h}| \cos\delta = h_1 \cos\delta = x_1 \tag{B.64}$$

$$\vec{i}_{\tilde{z}} \cdot \vec{h} = |\vec{i}_{\tilde{z}}||\vec{h}| \cos\theta = h_1 \cos\theta = z_\perp \tag{B.65}$$

Therefore:

$$x_1 = x_x x_h + y_x y_h + z_x z_h \tag{B.66}$$

$$z_\perp = x_z x_h + y_z y_h + z_z z_h \tag{B.67}$$

Finally,

$$x_2 = \Delta\tilde{x} - x_1 \tag{B.68}$$

In summary, the computation of the induced velocity contribution of a vortex sheet starts given the following: b/2R, γ_a, γ_b, $\vec{a}(x_a, y_a, z_a)$, $\vec{b}(x_b, y_b, z_b)$, z_m, and $\Delta\tilde{x}$. First the values of $|\vec{c}|$, c_a, and c_b are computed from Eqs. B.36, B.39, and B.40, respectively. Next, $\vec{i}_{\tilde{x}}$, $\vec{i}_{\tilde{y}}$, and $\vec{i}_{\tilde{z}}$ are expressed in the x, y, z coordinate system. The components of $\vec{i}_{\tilde{y}}(x_y, y_y, z_y)$ are computed from Eqs. B.45, B.46, and B.47, respectively. The components of $\vec{i}_{\tilde{x}}$ are computed in the order z_x, x_x, y_x from Eqs. B.48, B.50, and B.51, respectively. The components of $\vec{i}_{\tilde{z}}$ (x_z, y_z, z_z) are computed from Eqs. B.54, B.55, and B.56, respectively. Finally, the parameters x_1, x_2, and $z_\perp$ are computed from Eqs. B.66, B.68 and B.67, respectively.

Using the above results in Eq. B.22 or in Eq. B.23, the induced velocity ($\vec{q}_{VS}$) in the $\tilde{x}$, $\tilde{y}$, $\tilde{z}$ coordinate system can be computed. To obtain $\vec{q}_{VS}$ in the x, y, z coordinate system, the expressions developed above for $\vec{i}_{\tilde{x}}$ and $\vec{i}_{\tilde{z}}$ are substituted into Eq. B.22 or Eq. B.23. For example, substituting into Eq. B.23 gives:

$$\vec{q}_{vs} = \left(\frac{b}{2R}\right)\left(\frac{\gamma_a}{\Delta\tilde{x}}\right)\Big[\left(I_1 x_x + I_2 x_2\right)\vec{i} + \left(I_1 y_x + I_2 y_2\right)\vec{j}$$

$$+\left(I_1 z_x + I_2 z_2\right)\vec{k}\Big] \qquad (B.69)$$

B.3 Effect of a Viscous Vortex Core

Up to now the effects of a viscous core on a vortex sheet segment have been neglected. These effects are expected to be much smaller for a vortex sheet than for a vortex line because of the spread-out nature of the vortex sheet. However, these effects may still be important and should be investigated. To aid this investigation, the influence of a vortex core is developed for the case of a vortex sheet segment having constant circulation. The "fixed-wing vortex core model", discussed in Section 2.4, is used.

Consider Eq. B.2 specialized for the case of constant circulation ($\gamma_a = \gamma_b$):

$$\vec{q}_{vs} = \left(\frac{b}{2R}\right)\left(\frac{\gamma_a}{\Delta\tilde{x}}\right)\int_{x_2}^{x_1}\left(\frac{\cos\alpha + \cos\beta}{h}\right)\vec{n}\,d\tilde{x} \qquad (B.70)$$

Recall the vortex core circulation distribution from Eq. 2.5:

$$\gamma_c(h) = \frac{h^2}{h^2 + \rho_c^2} \qquad (B.71)$$

This $\gamma_c(h)$ was developed for vortex lines, but since the integral in Eq. B.70 is just the summing up of an infinite number of infinitesimal vortex lines $\gamma_c(h)$ applies here as well. Recall that $\gamma_c(h)$ is normalized by the total circulation (γ_a). From Eqs. B.3 and B.8:

$$\vec{h} = \frac{-z_\perp \vec{i}_{\tilde{x}} + \tilde{x}\,\vec{i}_{\tilde{z}}}{h} \qquad (B.72)$$

Now, putting $\gamma_c(h)$ inside the integral in Eq. B.70 and using Eq. B.72

$$\vec{q}_{vs} = \left(\frac{b}{2R}\right)\left(\frac{\gamma_a}{\Delta\tilde{x}}\right)\int_{x_2}^{x_1}\left(\frac{\cos\alpha + \cos\beta}{h^2 + \rho_c^2}\right)\left(-z_\perp \vec{i}_{\tilde{x}} + \tilde{x}\,\vec{i}_{\tilde{z}}\right)d\tilde{x} \tag{B.73}$$

Notice that the only difference which the addition of a vortex core makes is the change from h^2 in the denominator to $(h^2 + \rho_c^2)$. This causes a surprisingly large change in the integrated results because a fortunate cancelation no longer occurs.

The integrals I_1 and I_2 of Section B.1 are expanded into I_{1a}, I_{1b}, I_{2a}, and I_{2b} such that:

$$I_1 = I_{1a} + I_{1b} \tag{B.74}$$

$$I_2 = I_{2a} + I_{2b} \tag{B.75}$$

Recalling the definitions of cosα, cosβ, and h from Eqs. B.6, B.7, and B.3, respectively, the integrals are defined as follows:

$$I_{1a} = \int_{x_2}^{x_1}\left(\frac{c_a}{\sqrt{c_a^2 + \tilde{x}^2 + z_\perp^2}}\right)\left(\frac{-z_\perp\,d\tilde{x}}{\tilde{x}^2 + z_\perp^2 + \rho_c^2}\right) \tag{B.76}$$

$$I_{1b} = \int_{x_2}^{x_1}\left(\frac{c_b}{\sqrt{c_b^2 + \tilde{x}^2 + z_\perp^2}}\right)\left(\frac{-z_\perp\,d\tilde{x}}{\tilde{x}^2 + z_\perp^2 + \rho_c^2}\right) \tag{B.77}$$

$$I_{2a} = \int_{x_2}^{x_1}\left(\frac{c_a}{\sqrt{c_a^2 + \tilde{x}^2 + z_\perp^2}}\right)\left(\frac{\tilde{x}\,d\tilde{x}}{\tilde{x}^2 + z_\perp^2 + \rho_c^2}\right) \tag{B.78}$$

$$I_{2b} = \int_{x_2}^{x_1}\left(\frac{c_b}{\sqrt{c_b^2 + \tilde{x}^2 + z_\perp^2}}\right)\left(\frac{\tilde{x}\,d\tilde{x}}{\tilde{x}^2 + z_\perp^2 + \rho_c^2}\right) \tag{B.79}$$

There are two cases for each of these integrals $\rho_c^2 \le c_a^2, c_b^2$ and $\rho_c^2 \ge c_a^2, c_b^2$.

When $\rho_c = 0$, as in Section B.1, only the first case is possible. Recall the definitions of S_{1a}, S_{1b}, S_{2a} and S_{2b} from Eqs. B.14 through B.17, and define the following:

$$z_\rho \equiv \sqrt{z_\perp^2 + \rho_c^2} \tag{B.80}$$

$$C_{\rho a} \equiv \sqrt{C_a^2 - \rho_c^2} \tag{B.81}$$

$$C_{\rho b} \equiv \sqrt{C_b^2 - \rho_c^2} \tag{B.82}$$

$$R_{ca} \equiv \sqrt{\rho_c^2 - C_a^2} \tag{B.83}$$

$$R_{cb} \equiv \sqrt{\rho_c^2 - C_b^2} \tag{B.84}$$

The integrals can be evaluated as follows:

$$I_{1a} = \begin{cases} \left(\dfrac{C_a z_\perp}{z_a C_{\rho a}}\right)\left[\tan^{-1}\left(\dfrac{X_2 C_{\rho a}}{z_\rho S_{2a}}\right) - \tan^{-1}\left(\dfrac{X_1 C_{\rho a}}{z_\rho S_{1a}}\right)\right], & C_a^2 \geq \rho_c^2 \\ \left(\dfrac{C_a z_\perp}{2 z_\rho R_{ca}}\right) \ln\left[\dfrac{(z_\rho S_{1a} - X_1 R_{ca})(z_\rho S_{2a} + X_2 R_{ca})}{(z_\rho S_{1a} + X_1 R_{ca})(z_\rho S_{2a} - X_2 R_{ca})}\right], & \rho_c^2 \geq C_a^2 \end{cases} \tag{B.85}$$

$$I_{1b} = \begin{cases} \left(\dfrac{c_b z_\perp}{z_\rho C_{\rho b}}\right)\left[\tan^{-1}\left(\dfrac{x_2 C_{\rho b}}{z_\rho S_{2b}}\right) - \tan^{-1}\left(\dfrac{x_1 C_{\rho b}}{z_\rho S_{1b}}\right)\right], & c_b^2 \geqslant \rho_c^2 \\ \left(\dfrac{c_b z_\perp}{2 z_\rho C_{\rho b}}\right) \ln\left[\dfrac{(z_\rho S_{1b} - x_1 R_{cb})(z_\rho S_{2b} + x_2 R_{cb})}{(z_\rho S_{1b} + x_1 R_{cb})(z_\rho S_{2b} - x_2 R_{cb})}\right], & \rho_c^2 \geqslant c_b^2 \end{cases} \tag{B.86}$$

$$I_{2a} = \begin{cases} \left(\dfrac{c_a}{2 C_{\rho a}}\right) \ln\left[\dfrac{(S_{1a} - C_{\rho a})(S_{2a} + C_{\rho a})}{(S_{1a} + C_{\rho a})(S_{2a} - C_{\rho a})}\right], & c_a^2 \geqslant \rho_c^2 \\ \left(\dfrac{c_a}{R_{ca}}\right)\left[\tan^{-1}\left(\dfrac{S_{1a}}{R_{ca}}\right) - \tan^{-1}\left(\dfrac{S_{2a}}{R_{ca}}\right)\right], & \rho_c^2 \geqslant c_a^2 \end{cases} \tag{B.87}$$

$$I_{2b} = \begin{cases} \left(\dfrac{C_b}{2C_{\rho b}}\right) \ln\left[\dfrac{(S_{1b}-C_{\rho b})(S_{2b}+C_{\rho b})}{(S_{1b}+C_{\rho b})(S_{2b}-C_{\rho b})}\right] , & C_b^2 \geq \rho_c^2 \\ \left(\dfrac{C_b}{R_{cb}}\right)\left[\tan^{-1}\left(\dfrac{S_{1b}}{R_{cb}}\right) - \tan^{-1}\left(\dfrac{S_{2b}}{R_{cb}}\right)\right] , & \rho_c^2 \geq C_b^2 \end{cases}$$

(B.88)

Finally $\vec{q}_{VS}$ can be written as:

$$\vec{q}_{vs} = \left(\frac{b}{2R}\right)\left(\frac{\gamma_a}{\Delta\tilde{x}}\right)\left[\left(I_{1a}+I_{1b}\right)\vec{i}_{\tilde{x}} + \left(I_{2a}+I_{2b}\right)\vec{i}_{\tilde{z}}\right] \quad (B.89)$$

Notice that the integrals are either $0/\infty$ or $(\infty - \infty)/\infty$ when $c_a^2 = \rho_c^2$ or $c_b^2 = \rho_c^2$. To evaluate these cases only, I_{1a} and I_{2a} will be considered since I_{1b} and I_{2b} have the same behavior. First consider I_{1a} when $c_a^2 = \rho_c^2$, which implies that $C_{\rho a} = R_{ca} = 0$. To evaluate this case, use the following expansions for $\tan^{-1}$ and ℓn:

$$\tan^{-1}(x) = x + HOT \quad (B.90)$$

$$\ln(1+x) = x + HOT$$

(B.91)

Using these expansions on the $c_a^2 \geq \rho_c^2$ and the $\rho_c^2 \geq c_a^2$ expressions for I_{1a} gives the same answer in both cases, when $c_a^2 = \rho_c^2$:

$$I_{1a} = c_a z_\perp \left(\frac{x_2 S_{1a} - x_1 S_{2a}}{S_{1a} S_{2a} z_\rho^2}\right) \quad (B.92)$$

To evaluate I_{2a} when $c_a^2 = \rho_c^2$, a different expansion for $\tan^{-1}$ is used:

$$\tan^{-1}(x) = \frac{\pi}{2} - \frac{1}{x} + HOT \quad (B.93)$$

Again both the $c_a^2 \geq \rho_c^2$ and the $\rho_c^2 \geq c_a^2$ expressions give the same answer, when $c_a^2 = \rho_c^2$:

$$I_{2a} = \left(\frac{C_a}{S_{2a}} - \frac{C_a}{S_{1a}} \right) \tag{B.94}$$

APPENDIX C

VELOCITY INDUCED BY A CURVED VORTEX LINE SEGMENT

C.1 Computation of the Induced Velocity for a Given Arc Length and Radius of Curvature

Consider a circular arc vortex line segment of arc length $\Delta\theta_c$ and radius of curvature ρ_{VL} (nonD by R). Assume the vortex line has a viscous vortex core of radius ρ_c (nonD by R). Label the ends of the vortex line segment points P_η and P_a. Now consider the computation of the induced velocity at point P_η. Lamb in article 163 of [35] develops an expression for the self-induced velocity of a vortex ring. To solve the problem at hand, Lamb's expression can be used by subtracting the contribution from all of the ring except the segment between point P_η and P_a.

Assuming a circulation Γ and introducing different notation, Lamb's expression can be written:

$$q_{VR} = \frac{\Gamma}{4\pi R \rho_{VL}} \left[\ln\left(\frac{8\rho_{VL}}{\rho_c}\right) - \frac{1}{4} \right] \qquad \text{(C.1)}$$

It should be noted that Lamb developed this expression assuming $\rho_c << \rho_{VL}$. By symmetry, the induced velocity at point P_η due to half of the vortex ring starting at point P_η and ending 180° away is half of the result in Eq. C.1. This corresponds to the case $\Delta\theta_c = \pi(180°)$ for the curved vortex line segment problem. With circulation $\gamma(\gamma = \Gamma/2\pi b\Omega R)$, the induced velocity (nondimensional by tip speed) for the case $\Delta\theta_c = \pi$ can be written as:

$$q_{CVL}(\Delta\theta_c = \pi) = \gamma\left(\frac{b}{2R}\right)\left(\frac{1}{\rho_{VL}}\right)\left[\ln\left(\frac{8\rho_{VL}}{\rho_c}\right) - \frac{1}{4} \right] \qquad \text{(C.2)}$$

To find q_{CVL} for a general $\Delta\theta_c$, the induced velocity due to the part of the ring between $\theta_c = \Delta\theta_c$ and $\theta_c = \pi$ must be computed. From Eq. A.6 in Appendix A, the Biot Savart Relation can be written:

$$dq_{RVL} = \gamma\left(\frac{b}{2R}\right) \frac{\sin\theta \, ds}{r^2} \qquad \text{(C.3)}$$

where ds is the vortex line element at θ_c (nonD by R), r is the distance from point P_η to ds (nonD by R), θ is the angle between the tangent to the circular arc at ds and the line r, γ is the circulation at ds (nonD by $2\pi b\Omega R$), and b/2R is the rotor blade semichord over twice the rotor radius. In Fig. C.1 draw a tangent to the circular arc vortex line at ds. Drop a perpendicular of length h (nonD by R) from point P_η to the tangent. Now it can be seen that:

$$\sin\theta = \frac{h}{r} \tag{C.4}$$

In Fig. C.1 draw a radius of length ρ_{VL} (nonD by R) from the center of curvature to ds. Drop a perpendicular from point P_η to the radius. Now observe that the length of this perpendicular is $\rho_{VL} \sin \theta_c$. Further, observe that this perpendicular is parallel to the tangent which implies that:

$$h = \rho_{VL} - \rho_{VL} \cos\theta_c \tag{C.5}$$

Now from Fig. C.1:

$$r^2 = h^2 + \rho_{VL}^2 \sin^2\theta_c \tag{C.6}$$

Or, substituting for h:

$$r = \rho_{VL}\sqrt{2(1-\cos\theta_c)} \tag{C.7}$$

Substituting Eqs. C.5 and C.7 into Eq. C.4:

$$\sin\theta = \sqrt{\frac{1-\cos\theta_c}{2}} \tag{C.8}$$

Writing ds in terms of θ_c:

$$ds = \rho_{VL}\, d\theta_c \tag{C.9}$$

Substituting into Eq. C.3 and integrating from $\theta_c = \Delta\theta_c$ to $\theta_c = \pi$:

$$q_{RVL} = \gamma\left(\frac{b}{2R}\right)\int_{\Delta\theta_c}^{\pi}\left[\sqrt{\frac{1-\cos\theta_c}{2}}\;\frac{\rho_{VL}}{\rho_{VL}^2(2)(1-\cos\theta_c)}\right]d\theta_c \tag{C.10}$$

Recall the trig identity:

$$\sin\left(\frac{\theta_c}{2}\right) = \sqrt{\frac{1-\cos\theta_c}{2}} \tag{C.11}$$

Therefore:

$$q_{RVL} = \gamma\left(\frac{b}{2R}\right)\left(\frac{1}{4\rho_{VL}}\right)\int_{\Delta\phi}^{\pi}\frac{d\theta_c}{\sin(\theta_c/2)} \tag{C.12}$$

$$q_{RVL} = \gamma\left(\frac{b}{2R}\right)\left(\frac{1}{2\rho_{VL}}\right)\left[-\ln\tan\left(\frac{\Delta\theta_c}{4}\right)\right] \tag{C.13}$$

Now subtracting Eq. C.13 from Eq. C.2:

$$q_{CVL} = \gamma\left(\frac{b}{2R}\right)\left(\frac{1}{2\rho_{VL}}\right)\left\{\ln\left[\left(\frac{8\rho_{VL}}{\rho_c}\right)\tan\left(\frac{\Delta\theta_c}{4}\right)\right]-\frac{1}{4}\right\} \tag{C.14}$$

Equation C.14 is the desired expression for the induced velocity in terms of the arc length and the radius of curvature.

C.2 Determination of the Arc Length and Radius of Curvature

To determine $\Delta\theta_c$ and ρ_{VL}, the center of curvature must be found. Recall that three points determine a circle. On the tip vortex there is always at least one vortex line segment downstream of point P_η and at least one segment upstream. Label the ends of these adjacent vortex line segments points P_η and P_a and points P_b and P_η, respectively (Fig. C.2). Now the three points P_a, P_η, and P_b can be used to define a circular arc.

Define $\vec{a}_c$ and $\vec{b}_c$ as vectors (nonD by R) from point P_η to points P_a and P_b, respectively. Define a_c and b_c as the magnitudes of $\vec{a}_c$ and $\vec{b}_c$, respectively. Construct radii of length ρ_{VL} from the center of curvature to points P_η, P_a, and P_b (Fig. C.2). Define the angles α_c, β_c, $\Delta\theta_a$, and $\Delta\theta_b$ as shown in Fig. C.2. Drop perpendiculars from the center of curvature to $\vec{a}_c$ and $\vec{b}_c$ in Fig. C.2.

Notice that these perpendiculars are actually perpendicular bisectors since $\vec{a}_c$ and $\vec{b}_c$ are the bases of a pair of isosceles triangles having sides ρ_{VL} long. Therefore:

$$\rho_{VL} \cos \alpha_c = \frac{a_c}{2} \qquad \text{(C.15)}$$

$$\rho_{VL} \cos \beta_c = \frac{b_c}{2} \qquad \text{(C.16)}$$

$$\rho_{VL} \sin \alpha_c = \sqrt{\rho_{VL}^2 - \left(\frac{a_c}{2}\right)^2} \qquad \text{(C.17)}$$

$$\rho_{VL} \sin \beta_c = \sqrt{\rho_{VL}^2 - \left(\frac{b_c}{2}\right)^2} \qquad \text{(C.18)}$$

Recall the trig identity:

$$\cos\beta_c = \cos\left(\alpha_c + \beta_c - \alpha_c\right) = \cos(\alpha_c+\beta_c)\cos\alpha_c + \sin(\alpha_c+\beta_c)\sin\alpha_c \qquad \text{(C.19)}$$

Therefore, using Eqs. C.15, C.16, and C.17:

$$b_c = 2\rho_{VL}\cos\beta_c = a_c\cos(\alpha_c+\beta_c) + \sin(\alpha_c+\beta_c)\sqrt{4\rho_{VL}^2 - a_c^2} \qquad \text{(C.20)}$$

Solving for ρ_{VL}:

$$\rho_{VL} = \frac{1}{2}\sqrt{a_c^2 + \left[\frac{b_c - a_c\cos(\alpha_c+\beta_c)}{\sin(\alpha_c+\beta_c)}\right]^2} \qquad \text{(C.21)}$$

Recall the dot and cross product relations:

$$\vec{a}_c \cdot \vec{b}_c = a_c b_c \cos\left(\alpha_c + \beta_c\right) \qquad \text{(C.22)}$$

$$|\vec{a}_c \times \vec{b}_c| = a_c b_c \sin(\alpha_c+\beta_c) \qquad \text{(C.23)}$$

Now substituting into Eq. C.21:

$$\rho_{VL} = \frac{a_c}{2}\sqrt{1 + \frac{\left(b_c^2 - \vec{a}_c\cdot\vec{b}_c\right)^2}{|\vec{a}_c \times \vec{b}_c|^2}} \qquad \text{(C.24)}$$

Now it is necessary to find expressions for tan($\Delta\theta_a$/4) and tan($\Delta\theta_b$/4). Recall the trig identity:

$$\tan\left(\frac{\Delta\theta_a}{4}\right) = \frac{1-\cos\left(\frac{\Delta\theta_a}{2}\right)}{\sin\left(\frac{\Delta\theta_a}{2}\right)} \tag{C.25}$$

Observe in Fig. C.2 that:

$$\cos\left(\frac{\Delta\theta_a}{2}\right) = \sin\alpha_c \tag{C.26}$$

$$\sin\left(\frac{\Delta\theta_a}{2}\right) = \cos\alpha_c \tag{C.27}$$

Substituting into Eq. C.25 and multiplying by ($2\rho_{VL}/2\rho_{VL}$):

$$\tan\left(\frac{\Delta\theta_a}{4}\right) = \frac{2\rho_{VL} - 2\rho_{VL}\sin\alpha_c}{2\rho_{VL}\cos\alpha_c} \tag{C.28}$$

Substituting Eqs. C.15 and C.17 into Eq. C.28:

$$\tan\left(\frac{\Delta\theta_a}{4}\right) = \frac{2\rho_{VL} - \sqrt{4\rho_{VL}^2 - a_c^2}}{a_c} \tag{C.29}$$

By a similar process an expression can be developed for tan($\Delta\theta_b$/4):

$$\tan\left(\frac{\Delta\theta_b}{4}\right) = \frac{2\rho_{VL} - \sqrt{4\rho_{VL}^2 - b_c^2}}{b_c} \tag{C.30}$$

Define the ℓn expression in Eq. C.14 for $\Delta\theta_a$ and $\Delta\theta_b$ as:

$$AL \equiv \frac{8\rho_{VL}}{\rho_c}\tan\left(\frac{\Delta\theta_a}{4}\right) \tag{C.31}$$

$$BL \equiv \frac{8\rho_{VL}}{\rho_c}\tan\left(\frac{\Delta\theta_b}{4}\right) \tag{C.32}$$

Substituting Eqs. C.24 and C.30 into Eq. C.31:

$$AL = \frac{4a_c}{\rho_c|\vec{a}_c \times \vec{b}_c|^2}\left[|\vec{a}_c \times \vec{b}_c|^2 + (b_c^2 - \vec{a}_c \cdot \vec{b}_c)^2 - (b_c^2 - \vec{a}_c \cdot \vec{b}_c)\sqrt{|\vec{a}_c \times \vec{b}_c|^2 + (b_c^2 - \vec{a}_c \cdot \vec{b}_c)^2}\right] \tag{C.33}$$

To develop a similar expression for BL, note that an alternative expression for $_{VL}$ can be developed the same way Eq. C.24 was developed, but starting from:

$$\cos\alpha_c = \cos\left(\alpha_c + \beta_c - \beta_c\right) \tag{C.34}$$

instead of from Eq. C.19. The result is:

$$\rho_{VL} = \frac{b_c}{2}\sqrt{1 + \frac{\left(a_c^2 - \vec{a}_c\cdot\vec{b}_c\right)^2}{|\vec{a}_c\times\vec{b}_c|^2}} \tag{C.35}$$

Substituting Eqs. C.29 and C.35 into Eq. C.32:

$$BL = \frac{4\,b_c}{\rho_c|\vec{a}_c\times\vec{b}_c|^2}\left[|\vec{a}_c\times\vec{b}_c|^2 + \left(a_c^2 - \vec{a}_c\cdot\vec{b}_c\right)^2 - \left(a_c^2 - \vec{a}_c\cdot\vec{b}_c\right)\sqrt{|\vec{a}_c\times\vec{b}_c|^2 + \left(a_c^2 - \vec{a}_c\cdot\vec{b}_c\right)^2}\right] \tag{C.36}$$

To find the direction of the induced velocity $\vec{q}_{CVL}$, notice that points P_a, P_η and P_b form a plane and that the induced velocity must be perpendicular to this plane. Therefore, the induced velocity contributions of both $\Delta\theta_a$ and $\Delta\theta_b$ [$\vec{q}_{CVL}(\Delta\theta_a + \Delta\theta_b)$] are parallel to $\vec{a}_c \times \vec{b}_c$. Define the circulations (nonD by $2\pi b\Omega R$) of the segments $\Delta\theta_a$ and $\Delta\theta_b$ as γ_a and γ_b, respectively. The signs of γ_a and γ_b imply $\vec{q}_{CVL}(\Delta\theta_a + \Delta\theta_b)$ in the positive $\vec{a}_c \times \vec{b}_c$ direction (i.e., into the paper in Fig. C.2). Now multiplying $q_{CVL}(\Delta\theta_a + \Delta\theta_b)$ by $(\vec{a}_c \times \vec{b}_c)/(|\vec{a}_c \times \vec{b}_c|)$ to obtain $\vec{q}_{CVL}(\Delta\theta_a + \Delta\theta_b)$ and using Eqs. C.14, C.24, C.31, and C.32:

$$\vec{q}_{CVL}(\Delta\theta_a + \Delta\theta_b) = \left\{\frac{\gamma_a\left[\ln(AL) - 1/4\right] + \gamma_b\left[\ln(BL) - 1/4\right]}{a_c\sqrt{|\vec{a}_c\times\vec{b}_c|^2 + \left(b_c^2 - \vec{a}_c\cdot\vec{b}_c\right)^2}}\right\}\left(\frac{b}{2R}\right)\left(\vec{a}_c\times\vec{b}_c\right) \tag{C.37}$$

In summary, to compute $\vec{q}_{CVL}(\Delta\theta_a + \Delta\theta_b)$ given $\vec{a}_c$, $\vec{b}_c$, γ_a, γ_b, ρ_c, and b/2R, use Eqs. C.33, C.36, and C.37.

APPENDIX D

TIP VORTEX CORE MODELS

D.1 Introduction

The simple solid body rotation vortex core model is adequately described in Section 2.4. This appendix documents the two more complex vortex core models in more detail than is appropriate for Section 2.4. In addition, the Betz theory of wing-wake roll-up is applied to both the fixed-wing and rotary-wing cases.

D.2 Fixed-Wing Vortex Core Model

The model normally used for all straight wake elements is the "fixed-wing vortex core model" (FWVCM). This model is based on measurements of the properties of tip vortices in the wake of fixed-wing and wind-tunnel models [36,37,38, 43]. The data from these references, which is used here, covers rectangular planform wings with aspect ratios from 4.2 to 6, Reynolds numbers from 3×10^5 to 7×10^6, and Mach numbers from 0.08 to 0.5. The data given in the references often covers a range of distances downstream from the wing. When it is possible, the data used here is for downstream distances (or times) which are typical of the first blade-vortex interaction on multibladed helicopter rotors.

The data in [36,37,38,43] contains some plots of tangential induced velocity (V_θ) vs. radial distance from the vortex centerline ρ. The primary data, however, is peak tangential induced velocity V_θ^* and the radius at which this V_θ^* occurs. This radius is, by definition, the vortex-core radius (ρ_c, when normalized by the wing span R). The first step in constructing a vortex-core model is to find expressions for V_θ^* and ρ_c. Due to the limited range of aspect ratio, it is not possible to decide from the data whether ρ_c is a function of wing span R or wing chord c. Intuitively the chord seems more reasonable. On this basis, ρ_c varies from 2.5% of the chord to 6.3% of the chord in the data selected from [36, 37, 38, 43]. A value of 5% of the chord is, therefore used for ρ_c in the standard FWVCM; however, ρ_c is a parameter which can be varied to suit various theories.

The peak velocity V_θ^* can be normalized by $\Gamma/(4\pi R\rho_c)$, where Γ is the tip vortex circulation. The problem with this normalization of V_θ^* is that Γ is difficult to measure accurately. A much easier quantity to measure is the lift coefficient C_L. Assuming that the tip vortex circulation Γ is equal to the average bound circulation:

$$\rho V \Gamma = \frac{1}{2}\rho V^2 c C_L \tag{D.1}$$

where ρ = air density, V = wind speed, and c = wing chord. This allows the normalizing factor to be written in terms of C_L:

$$\frac{\Gamma}{4\pi R \rho_c} = \frac{c V C_L}{8\pi R \rho_c} \tag{D.2}$$

In Section D.4, a theory is presented which says that Γ should be equal to the maximum bound circulation instead of the average. For a typical rectangular wing case this increases the normalizing factor based on C_L by about 7%, which is not significant, given the scatter in the data reviewed here.

Using either the normalizing factor based on Γ or based on C_L, as appropriate, selected V_θ^* data has been normalized. The resulting values range from 0.84 to 1.39 or from 0.78 to 1.30, depending upon whether the average Γ or the peak Γ is used. A value of 1.0 is chosen as representative and convenient. Therefore:

$$V_\theta^* = \frac{\Gamma}{4\pi R \rho_c} \tag{D.3}$$

The peak velocity V_θ^* and the location of the peak ρ_c have been determined. The next step is to fit a smooth curve through this peak which fairs into Biot Savart for large ρ and which is linear for small ρ. A simple function which does this is:

$$V_\theta = \frac{\Gamma}{4\pi R \rho_c}\left(\frac{2\rho}{1+\rho^2}\right) \tag{D.4}$$

This implies a radial circulation distribution:

$$\gamma_c(\rho) = \left(\frac{\rho^2}{1+\rho^2}\right) \tag{D.5}$$

In Fig. D.1, this FWVCM function (Eq. D.4) is compared with measured data from Fig. 6 of [43] and with Biot Savart. For this plot, the measured values of vortex core radius (5.3% chord) and normalized V_θ^* (0.968) are used, instead of the standard values (5% and 1.0 , respectively). This makes the peaks match between the measured data and the FWVCM,which allows a direct comparison of the shape of the curves.

The FWVCM function can be seen to approach Biot Savart faster than the measured data. A more complicated function could be found to improve this; however, given the uncertainty in ρ_c and V_θ^* , it is not worth the extra expense. The measured data is above Biot Savart for large ρ, in Fig. D.1. This implies that the Γ used is too small. This Γ is reduced by a factor of 0.968 from the Γ of Eq. D.1 in order to match peak heights. Furthermore, as mentioned above, the Γ of Eq. D.1 may be 7% too small. Finally, the measured curve in Fig. D.1 is a mean drawn through the actual data points and may not be completely accurate.

In summary, the fixed-wing vortex core model is a very simple model based on fixed-wing tip vortex measurements. It is much more realistic than the solid-body rotation model which has an induced velocity peak twice as high. Finally, it is about as complicated a model as can be justified, given the variations in the measured data and the lack of understanding of these variations.

D.3 Rotary-Winged Vortex Core Model

Recently, detailed measurements of tip vortex properties for a rotor at full-scale Mach and Reynolds numbers were published in [39]. This data shows a vortex core radius almost an order of magnitude smaller than the fixed-wing data, with a peak induced velocity which is nearly the same. This implies a very different tip vortex radial circulation distribution for rotors compared to fixed wings. A different vortex core model, the "rotary-wing vortex core model" (RWVCM), is used to represent this data.

The data in [39] is presented as a function of the age of the tip vortex measured in degrees of azimuth angle. Since we are primarily concerned with the properties of the tip vortex when it first interacts with the following blade, only the data for an age of about 100° is considered. This data shows an average vortex-core radius (ρ_c) of about 0.008 chords and an average peak tangential induced velocity (V_θ^*) of about $0.20(\Gamma/4\pi R\rho_c)$, where ρ_c is normalized by R. A typical plot of tangential induced velocity vs. radius is given in Fig. 13 of [39]. A function which approximates this measured data outboard of the vortex core is:

$$V_\theta = \frac{V_\theta^*}{\rho}\left[10 - 9\, e^{-.075(\rho-1)}\right] \tag{D.6}$$

where ρ is normalized by the vortex core radius ρ_c. This function is compared with the measured data and with the Biot-Savart relation in Fig. D.2. A function with a maximum at $\rho = 1$ would be desirable but the function of Eq. D.6 is satisfactory. The induced velocity inside the vortex core can be represented by Eq. D.5 multiplied by 0.2 to get the correct peak height. This gives the following expression for V_θ:

$$V_\theta = \begin{cases} \dfrac{\Gamma}{4\pi R\rho_c}\left(\dfrac{0.4\rho}{1+\rho^2}\right) & , \ \rho \leq 1.0 \\[2ex] \dfrac{\Gamma}{4\pi R\rho_c}\left(\dfrac{2}{\rho}\right)\left[1.0 - 0.9\, e^{-.075(\rho-1)}\right] & , \ \rho \geq 1.0 \end{cases} \tag{D.7}$$

Using this expression for V_θ the radial distribution of vortex core circulation $\gamma_c(\rho)$ can be determined.

$$\gamma_c(\rho) = \begin{cases} \left(\dfrac{0.2\rho^2}{1+\rho^2}\right) & , \ \rho \leq 1.0 \\[2ex] \left[1.0 - 0.9\, e^{-.075(\rho-1)}\right] & , \ \rho \geq = 1.0 \end{cases} \tag{D.8}$$

Recently, some preliminary measurements of model rotor tip vortex properties have been published [44]. These measurements show tip vortex properties closer to the fixed-wing data [36,37,38] than to the rotor-wing data [39]. In addition, [44] refers to [45], which contains measurements of tip vortex properties for a full-scale rotor. The measurements in [45] are also closer to the fixed-wing measurements. This casts some doubt on the validity of the RWVCM, which is another reason the RWVCM remains experimental while the FWVCM is standard.

D.4 Betz Theory for Tip Vortex Roll-Up

Betz [41] has developed a theory for the roll-up of the trailing vortex wake of a wing. This theory has been simplified and extended by Donaldson, Rossow, et al.[46,47,48]. The net result is a method for computing the radial distribution of circulation in the final rolled-up wake [$\Gamma_c(r)$], given the spanwise distribution of bound circulation on the wing [$\Gamma_b(y)$]. Define the centroid of all bound circulation outboard of any spanwise station y as $\bar{y}$:

$$\bar{y} \equiv \frac{\int_y^1 \frac{d\Gamma_b(\xi)}{d\xi}\,\xi\,d\xi}{-\Gamma_b(y)} \qquad \text{(D.9)}$$

where y and $\bar{y}$ are normalized by the span. Then let the radial coordinate of the rolled-up vortex (normalized by span) be r. The theory says that when:

$$r = \bar{y} - y \qquad \text{(D.10)}$$

then:

$$\Gamma_c(r) = \Gamma_b(y) \qquad \text{(D.11)}$$

In other words, the total circulation within a radius r of the center of the rolled-up vortex core [$\Gamma_c(r)$] is the same as the bound circulation on the wing at spanwise station y [$\Gamma_b(y)$], provided that Eq. D.10 is satisfied.

As is written above, the theory applies to the roll-up into a tip vortex of the wake outboard of spanwise station y_m on the wing where the magnitude of the slope of the bound circulation $|d\Gamma_b(y)/dy|$ decreases monotonically from a maximum at the wing tip to a minimum at y_m. The theory for more complicated $\Gamma_b(y)$ functions and for inboard wake roll-up is given by Donaldson [46]. Since this is an inviscid theory, it cannot be expected to predict a viscous phenomenon like the vortex core radius. Indeed, given the required nature of $\Gamma_b(y)$, the induced velocity will rise monotonically to a peak as r goes from

$+\infty$ to 0+. It will then jump to an equal peak of opposite sign at r = 0- and decrease monotonically as r goes to $-\infty$. The magnitude of the induced velocity peak is related to the slope of $\Gamma_b(y)$ at the wing tip and has nothing to do with viscosity. Thus, this theory cannot be expected to predict either the vortex core radius ρ_c or the magnitude of the peak tangentially induced velocity V^*_θ. The theory may be able to predict the reduction of the measured tangentially induced velocity V_θ below that predicted by Biot Savart for some of the tip vortex outside of the vortex core.

To apply this theory to rotors, replace the word wing with rotor blade, replace the word spanwise with radial, replace y and y_m with η and η_m, and consider an appropriate $\Gamma_b(\eta)$. An obvious difference between wings and rotors is the location of y_m. For simply loaded wings, the minimum $|d\Gamma_b(y)/dy|$ would be expected to correspond with the maximum $\Gamma_b(y)$ and be at the center of the wing (y_m = 0.5). For rotors, the average around the azimuth of the minimum $|d\Gamma_b(\eta)/d\eta|$ also corresponds to the maximum $\Gamma_b(\eta)$, but this maximum is located outboard of the middle of the rotor blade, typically η_m = 0.85 - 0.95.

To simplify the computations, define a coordiante x which runs from x = 1.0 at $\eta = \eta_m$ to x = 0.0 at η = 1.0, the blade tip. This means that:

$$x(1-\eta_m) = 1-\eta \qquad \text{(D.12)}$$

Define a new radial coordinate from the center of the tip vortex normalized by $(1 - \eta_m)R$ instead of R:

$$\rho_x \equiv r/(1-\eta_m) \qquad \text{(D.13)}$$

Define a new bound circulation distribution $\gamma_b(x)$ and a new radial tip vortex circulation distribution $\gamma_c(\rho_x)$, both normalized by the total tip vortex circulation Γ.

$$\gamma_b(x) \equiv \Gamma_b(\eta)/\Gamma \qquad \text{(D.14)}$$

$$\gamma_c(\rho_x) \equiv \Gamma_c(r)/\Gamma \qquad \text{(D.15)}$$

Note that Γ will be equal to the $\Gamma_b(\eta)$ when $\eta = \eta_m$.

$$\Gamma = \Gamma_b(\eta_m) \tag{D.16}$$

Now introducing the new variables and combining Eqs. D.9 and D.10:

$$\rho_x(x) = \frac{x\gamma_b(x) + \int_x^0 \frac{d\gamma_b(\xi)}{d\xi}\,\xi\,d\xi}{\gamma_b(x)} \tag{D.17}$$

The Biot Savart relation gives the tangential induced velocity V_θ as a function of radius r as:

$$V_\theta = \frac{\Gamma_c(r)}{2\pi R r} \tag{D.18}$$

Introducing the new notation:

$$V_\theta = \frac{\Gamma}{2\pi R(1-\eta_m)} \quad \frac{\gamma_c(\rho_x)}{\rho_x(x)} \tag{D.19}$$

Recalling Eqs. D.11, D.14, and D.15:

$$V_\theta = \frac{\Gamma}{2\pi R(1-\eta_m)} \quad \frac{\gamma_b(x)}{\rho_x(x)} \tag{D.20}$$

If, for a given $\gamma_b(x)$, one is able to invert $\rho_x(x)$ to obtain $x(\rho_x)$, then $\gamma_c(\rho_x)$ can be found by substitution of $x(\rho_x)$ into $\gamma_b(x)$. Normally, this is not practical and Eq. D.20 must be used. Equation D.20 gives $V_\theta(x)$, not $V_\theta(\rho_x)$. To obtain $V_\theta(\rho_x)$, the functions $V_\theta(x)$ and $\rho_x(x)$ must be tabulated or cross-plotted. Finally, $V_\theta(r)$ is obtained by the use of Eq. D.13.

Define a normalized tangentially induced velocity $v(\rho_x)$:

$$v(\rho_x) \equiv \frac{\gamma_b(x)}{\rho_x(x)} \tag{D.21}$$

This $v(\rho_x)$ can be plotted vs. $\rho_x(x)$ to obtain a normalized version of the V_θ vs. r plot. The interesting thing about this is that all of the effects of changes in $\gamma_b(x)$ are in $v(\rho_x)$, while all of the effects of changes in η_m are in the normalizing factors for $v\ (\rho_x)$ and $\rho_x(x)$.

For an ideal, concentrated vortex Biot Savart gives:

$$v(\rho_x) = 1.0 / \rho_x(x) \tag{D.22}$$

By definition: $\gamma_b(x) = 1.0$ for $x \geq 1.0$; therefore the Biot Savart value of $v(x)$ applies for all values of $\rho_x(x) > \rho_x(1.0)$, no matter what $\gamma_b(x)$ is used. For the classic elliptical spanwise loading:

$$\gamma_b(\eta) = \sqrt{1-\eta^2} \tag{D.23}$$

which implies:

$$\gamma_b(x) = \sqrt{2x - x^2} \tag{D.24}$$

This, in turn, gives:

$$\rho_x(x) = \frac{\frac{\pi}{2} - \sin^{-1}(1-x)}{2\sqrt{2x-x^2}} - \left(\frac{1-x}{2}\right) \tag{D.25}$$

Another $\gamma_b(x)$ which might be used to approximate the spanwise loading on a rectangular wing is:

$$\gamma_b(x) = (1-\delta) + \delta\sqrt{2x - x^2} \tag{D.26}$$

This gives:

$$\rho_x(x) = \frac{\frac{\delta\pi}{2} - \delta\sin^{-1}(1-x) + (1-\delta)(1+x)}{2\sqrt{2x-x^2}} - \left(\frac{1-x}{2}\right) \tag{D.27}$$

Figure D.3 shows the $v(\rho_x)$ vs. $\rho_x(x)$ plots for these three $\gamma_b(x)$ (Eqs. D.22, D.24, and D.26). The value of $\delta = 0.4$ used in Eq. D.26 represents a rectangular wing as will be discussed later.

Now consider the effect of going from $\eta_m = 0.5$ (fixed wing) to $\eta_m = 0.95$ (rotary wing in hover). Using Eq. D.13 for a given r: for $\eta_m = 0.5$, $\rho_x = 2r$ and for $\eta_m = 0.95$, $\rho_x = 20r$. From Fig. D.3 we see that v(x) increases as $\rho_x(x)$ decreases. Thus, $v(\rho_x)$ increases as η_m decreases, for a given r and $\gamma_b(x)$. Let $V_\theta = K\, v(\rho_x)$ for $\eta_m = 0.50$, where K is a constant. Then, for the same r and $\gamma_b(x)$, $V_\theta = 10K\, v(10\rho_x)$ for $\eta_m = 0.95$. The question is: which is greater $Kv(\rho_x)$ or $10Kv(10\rho_x)$? Consider the case when they are equal:

$$K\, v(\rho_x) = 10\, K\, v(10\rho_x) \tag{D.28}$$

It is easily shown that Eq. D.22 (the Biot Savart case) satisfies Eq. D.28. Clearly, for any $v(\rho_x)$ which rises less rapidly with decreasing ρ_x, Eq. D.22 will result in:

$$K\, v(\rho_x) < 10\, K\, v(10\rho_x) \tag{D.29}$$

For the type of $\gamma_b(x)$ required by this restricted version of the theory, the $\gamma_c(\rho_x)$ are all monotonically decreasing functions going from $\gamma_c(\rho_x(1.0)) = 1.0$ to $\gamma_c(0.0) = 0.0$. Thus $v(\rho_x)$ is always less steep than Eq. D.22 for $\rho_x(x) < \rho_x(1.0)$, and Eq. D.29 applies. In general, therefore, if $\gamma_b(x)$ is fixed and η_m is allowed to change, then at any given $r < (1 - \eta_m)\, \rho_x(1.0)$, V_θ will increase as η_m increases.

Flight test data [49,50,51] shows that the radial peak of the zeroth harmonic (azimuthal average) of the bound circulation on a helicopter rotor blade varies from about $\eta_m = 0.85$ at moderate advance ratios to about $\eta_m = 0.95$ at hover. Data was only taken at a limited number of radial stations in these tests: $\eta = 0.25, 0.40, 0.55, 0.75, 0.85, 0.90, 0.95$ for [49,50] and $\eta = 0.40, 0.75, 0.85, 0.95, 0.98$ for [51]. This limited data is enough to estimate η_m but not enough to define the shape of $\gamma_b(x)$.

For fixed wings, Fig. 3 of [42] gives the spanwise airload distribution for one of the cases later repeated in [37] and quoted here in Table I. For this case $\eta_m = 0.7$, assuming the oscillations near the wing tip are faired out, as is done in [48]. Fairing out these oscillations near the tip amounts to assuming that the various independent vortices generated by these oscillations will ultimately all rollup into one main vortex. Betz theory ignores this possibility, so the only way to achieve this final result is to eliminate the oscillations.

Looking back at Fig. 8, we see that V_θ from the FWVCM is greater than the V_θ from the RWVCM for $\rho > 1.0$. This area outside of the core ($\rho > 1.0$) is where the Betz model should work. The RWVCM is based on hovering rotor data, and from the discussion above, should have $\eta_m = 0.95$. Similarly, the FWVCM data should have an η_m between 0.5 and 0.7. Thus, V_θ is increasing while η_m is decreasing. This implies that the $\gamma_b(x)$ must be very much more tip-loaded for the fixed-wing case compared to the rotary-wing case in order to overcome the effect of η_m on V_θ. Another possibility is that the oscillations in the fixed-wing data should not be faired, in which case $\eta_m = 0.98$. This will be examined below.

For the rotary-wing case, the measured data of Fig. 13 in [39] is plotted in Fig. D.4. For comparison, the Biot Savart value of V_θ (also from [39]) and the Betz model value of V_θ for an elliptical $\gamma_b(x)$ are also plotted. The agreement is reasonable but the use of an elliptical $\gamma_b(x)$ is questionable. The data in [51] shows that the bound circulation at $\eta = 0.98$ is about 92% of the bound circulation at $\eta = 0.95$. The elliptical $\gamma_b(x)$ gives only 80% at $\eta = 0.98$. If $\gamma_b(x)$ from Eq. D.26 is used, a $\delta = 0.4$ is needed to get the 92%. The $v(\rho_x)$ for this $\gamma_b(x)$ is compared with Biot Savart and with an elliptical $\gamma_b(x)$ in Fig. D.3. Comparing this plot with Fig. D.4, it is clear that this $\delta = 0.4$ case is much too close to Biot Savart to give good agreement with the measured data. Thus, the correlation of the Betz theory with these rotary-wing measurements is not completely satisfactory. For a real test of the theory, simultaneous measurement of the bound circulation and the tip vortex properties is needed. This is not currently available for the rotary-wing case.

For the fixed-wing case, the measured data from Fig. 9 of [43] is presented in Fig. D.5. This data was taken using the same wing as [42]. Based on the span-loading from [42], an $\eta_m = 0.7$ and a $\delta = 0.3$ is chosen. Therefore

$$\gamma_b(x) = 0.7 + 0.3\sqrt{2x - x^2} \qquad (D.30)$$

In this case, α is known, $\partial C_L/\partial\alpha = 4.0$ is assumed, and Γ is computed, using Eq. D.1. The resulting V_θ for both the $\gamma_b(x)$ of Eq. D.30 and for Biot Savart are plotted in Fig. D.5. There is reasonable agreement out to $r/(b/2) = 0.15$, where the measured data levels out. If the oscillations in the span-loading (near the wing tip) are not faired out, then $\eta_m = 0.98$. For this large η_m, $\rho_x(1.0)$ corresponds to $r/(b/2) < 0.04$ for either the elliptical or the rectangular $\gamma_b(x)$. Since $\gamma_b(x) = 1.0$ for all $x > 1.0$, this means that $\eta_m = 0.98$ gives the Biot Savart value for V_θ for all $r/(b/2) > 0.04$. Figure D.5 shows that this results in much worse agreement with the measured data when compared with the $\eta_m = 0.7$ case (Betz Rectangular). Thus, the fairing seems to be justified.

In summary, the Betz model seems to work fairly well for the fixed-wing case, provided the oscillations are faired out. More experimental data (especially on span-loadings near the tip) is needed to verify the Betz model for rotary wings. The most practical result from the Betz model is the location of η_m at $|d\Gamma_b/d\eta|_{\text{min.}}$, which is normally at the maximum $\Gamma_b(\eta)$. This gives a theoretical foundation for using the peak-bound circulation (instead of the average) as the total circulation of the rolled-up tip vortex.

APPENDIX E

AIRLOADS PROGRAM LDS-73

A listing of LDS-73 is provided in Appendix F. This appendix supplements that listing. An explanation of the order of the input data cards is provided. Most of the data is input by name lists. A set of lists defining all of the inputs in each name list is provided. Many of these input variables also appear in a COMMON statement and this is noted. A sample set of input data is given together with some notes on typical values for the inputs. A discussion of the output from LDS-73 is also provided. The output from a sample case is not provided, due to its great bulk.

LDS-73 is broken up into many subroutines. A short description of what each subroutine does is provided by comment cards in the listing in Appendix F. Most of the important variables in LDS-73 are transmitted between subroutines by various labeled COMMON statements. A set of lists defining all of the variables in each of the COMMON statements is provided. If a variable also appears in an input name list, this is noted. More detailed definitions of some of these variables appear in the List of Symbols.

E.1 LDS-73 Input Data

1) Read GENDAT (real) at start of each case
2) Read OPTCON (integer) at start of each case
3) If (ND>0 and ND≤3) read distortion $\overleftarrow{D}(\psi,\delta)$ as punched out by WG-71

```
        DO K=1, NPHI
   10  READ 20, ((D(I,J,K),I=1,3), J=1, NPSI)
   20  FORMAT (6E13.6)
```

4) $\left.\begin{array}{l}\text{If (IPLX>0)}\\ \text{and (IRL}\neq\text{2)}\end{array}\right\}$ Read experimental airloads XLDS(η,ψ)

```
                DO  10  K=1, NE
   10  READ 20, (XLDS(K,J), J=1, NPSI)
   20  FORMAT (10F7.3)
```

5) If (IG=1) Read $\alpha(\eta,\psi)$ from previous iteration plus $\eta,\Delta\eta$

```
              DO  10  K=1, NE
      10  READ 20, (ALPHA(K,J), J=1,NPSI)
          READ 30, (ETG(K), K=1,NE)
          READ 30, (DETA(K), K=1,NE)
      20  FORMAT (5E15.7)
      30  FORMAT (10F6.3)
```

6) If (IRL$\neq$2) Read ROTOR

7) If (IRB<2) Read BEND

8) If (NG>IPRF) Read PERF

```
NAMELIST/PERF/IPRF,TH1,TC,TW
```

These inputs are defined under GENDAT(TH1), OPTCOU(IPRF), and ROTOR(TC,TW)

Sample Cases

A) Rigid wake (ND=0), start with $\gamma_{bv}(\phi) = \gamma_o$(IG=2), plot experimental airloads (IPLX=1):

GENDAT
OPTCOU
XLDS(η,ψ) cards
ROTOR
BEND

Note: Controls are set by program so that $\vec{D}$, XLDS, ALPHA, ROTOR, BEND are only read on the first iteration. When OPTCOU is read again at the start of a new case, these controls can be reset to read in these inputs again if desired.

B) Distorted tip vortex (ND=3), input $\alpha(\eta,\psi)$(IG=1), no XLDS(IPLX=0)

GENDAT
OPTCOU
$\vec{D}(\psi,\delta)$ cards
$\alpha(\eta,\psi)$ cards
η card
$\Delta\eta$ card
ROTOR
BEND

Note: If it is desired to follow Case B with another case using the same $\vec{D}$, then when OPTCOU is read in again, make ND=4. This avoids reading in $\vec{D}$ again and the program will reset ND=3 before execution. If a different $\vec{D}$ is to be read in, leave ND=3.

C) In Case B, assume it is desired to try two different rotor blad designs (twist or thickness/chord). Then make IPRF<NI and read PERF after BEND.

E.2 Name Lists

E.2.1 NAMELIST/GENDAT/

EMU	μ(COMMON/SGAM) - Advance ratio ($V/\Omega R$)
AMBDA	λ(COMMON/SQLDS) - Uniform, average downwash perpendicular to TPP - used to determine rigid wake geometry. (nonD by tip speed)
EM	m - number of turns of wake included in the wake model - azimuthal extent of wake = $2\pi m$
RHOK	ρ_k(COMMON/SQLDS) - Nonburst value of ρ_c - vortex core radius. (nonD by R)
DPS1	$\Delta\psi$ - Azimuth angle interval between positions of η blade on which airloads are computed (degrees) - Note: DANG=$\Delta\psi$ (radians)
AO	a_o (COMMON/BURST) - Coning angle (zeroth harmonic of rigid body flapping). This input value of a_o is used only to compute $\vec{a}$ and $\vec{b}$ from P_η to P_a and P_b. The value of a_o computed in the blade motion portion of LDS-73 is a separate variable (radians).
BOER	b/2R (COMMON/SGAM) - b = rotor blade semichord (ft), R = rotor radius (ft)
TANMU	$\mu\tan(i)$ (COMMON/SQLDS) - Inflow (downwash) perpendicular to TPP due to flight speed (relative wind). (nonD by tip speed)
GZER	γ_o(COMMON/SBVGS) - Constant value of bound circulation used for zeroth iteration (IG>1). (nonD by $2\pi b\ \Omega R$)
EPSS	ε_s - Distance behind η blade to start of near shed wake vortex sheet (when IEPS$\neq$2). (nonD by b)

CTO	C_T (COMMON/SLDS) - Input target value of thrust coefficient, iteration on collective pitch (θ_o) until C_T is achieved. $C_T = \dfrac{\text{Thrust}}{\rho\pi R^2 (\Omega R)^2}$
SM	σ(COMMON/SLDS) - Rotor solidity $\sigma = \dfrac{\text{rotor blade area}}{\text{rotor disk area}} = \dfrac{n_b C}{\pi R}$
A	a (COMMON/SGAM) - $\partial C_2/\partial\alpha$ - Input lift curve slope which is a constant. Various factors are used in blade motion and airloads computations to modify the effective $\partial C_L/\partial\alpha$ (per radian).
TH1	θ_1 (COMMON/SLDS) - Rotor blade linear twist (radians), $\theta(\eta) = \theta_o + \theta_1\eta$.
RHOG	ρ_g (COMMON/BURST)- Burst vortex core radius. (nonD by R).
WF	w_f (COMMON/SGAM) - Weighting factor, to start a new iteration for circulation or blade motion take w_f times latest results plus $(1 - w_f)$ times previous results.
VT	$V_T = \Omega R$ (COMMON/SLDS) - Rotor tip speed (ft/sec)
DEL	(COMMON/PLT) - Interval size on airloads plots ordinate (lb/in)
ELMN(6)	(COMMON/PLT) - Minimum value on airloads plot ordinate for each different η (lb/in)
QMO	q_{mo} (COMMON/SQVLS) - Minimum value of the induced velocity contribution of any vortex line or vortex sheet segment for which a special printout is made in OSVL or OVS (nonD by tip speed).
ETG(6)	η_k (COMMON/SGAM) - Radial stations on η blade of point P_η at which airloads are computed (nonD by R).
DELTA(6)	$\Delta\eta_k$ (COMMON/SGAM) - Delta η_k used in radial integrations (nonD by R).

EL(3)	ℓ_ℓ (COMMON/SBVGS) - Radial stations on ℓ blade of points P_ℓ from which elements of vortex wake are shed. ℓ_1 = inboard edge of inboard trailing vortex sheet (ITVS) ℓ_2 = radial station of inboard trailing vortex line ℓ_3 = tip vortex (normally $\ell_3 = 1.0$) (NonD by R)
DVB	(COMMON/BURST) - If $\Delta\ell_{xy} <$ DVB then delay tip vortex core bursting by $[1 + 2(\text{DVB}-\Delta\ell_{xy})]$ (truncated to an integer) $\Delta\phi$ intervals. This controls how much of any tip vortex line segment must have passed close to the η blade before the entire segment is burst.
FVB	(COMMON/BURST) - If a tip vortex line segment passes within (FVB)ρ_c above or below the η blade and the nearest end of the segment is between one segment length behind the η blade and one-half segment length ahead of the η blade ($1.0 \leq \Delta\ell_{xy} \leq -1.5$) then a close blade-tip vortex interaction has occurred.
ELI(25)	$\ell_i(\phi)$ (COMMON/SQLDS) - Radial station of outboard edge of inboard trailing vortex sheet at azimuth angle ϕ(nonD by R)
ELT(25)	$\ell_t(\phi)$ (COMMON/SQLDS) - Radial station of inboard edge of tip vortex sheet at azimuth angle ϕ (nonD by R)
FTO(25)	$f_t^o(\phi)$ (COMMON/SQLDS) - Fraction of total tip vortex circulation concentrated into tip vortex line at azimuth angle ϕ and zero age ($\delta=0$)
FTD	f_t^D (COMMON/SQLDS) - Tip vortex rollup rate. Fraction of total tip vortex circulation concentrated into tip vortex line = $f_t(\phi,\delta)$ $f_t(\phi,\delta) = f_t^o(\phi) + (f_t^D)\delta$ until $f_t(\phi,\delta)$ reaches 1.0 then $f_t(\phi,\delta) = 1.0$.
VSM	(COMMON/SQVLS) - Vortex sheet margin. If necessary the length ($\Delta\ell$) of a vortex sheet segment is extended

so that the projection of P_η onto the vortex sheet is never closer than (VSM)$\Delta\ell$ to either the front or back edges of the segment.

RMVS	ρ_{vs} (COMMON/SQVLS) - Vortex sheet core radius (nonD by R)
DAM	$\Delta\alpha_m$ (COMMON/SQVLS) - Maximum change in local angle of attack due to any one vortex line or sheet segment (radians)

E.2.2 NAMELIST/OPTCOU

NB	n_b (COMMON/SQLDS) - Number of rotor blades
NE	n_η (COMMON/SGAM) - Number of radial stations η_k on η blade at which airloads are computed
NL	n_ℓ (COMMON/ SQLDS) - Number of radial stations on ℓ blade. For LDS-73, NL = 3.
NC	Number of cases - When a case is finished NC is decreased by one and if NC is then greater than zero GENDAT and OPTCOU are read in and a new case is started.
NI	(COMMON/SLDS) - Number of circulation iterations, when NG = NI this case is finished.
NG	(COMMON/SGAM) - Current iteration number, increased by one after each iteration
IG	(COMMON/SBVGS) - Indicates type of circulation computation used on first iteration. IG=0: Compute bound circulation $[\gamma_{bv}(\psi)]$ using angle of attack $\alpha(\eta,\psi)$ from previous iteration stored in ALPHA(6,25) IG=1: Compute $\gamma_{bv}(\psi)$ using $\alpha(\eta,\psi)$ read in from cards along with associated η_k and $\Delta\eta_k$ which can be different from η_k and $\Delta\eta_k$ in the rest of the program

	IG>1: Zeroth iteration (NG=0), set all $\gamma_{bv}(\psi) = \gamma_o$ = GZER (input GENDAT) Note: $\gamma_{bv}(\psi)$ = constant implies no shed wake; however, γ_s is made very small (but not zero) so that it will make a negligible contribution to downwash, while the influence coefficient matrix QMAT can still be computed for use in later iterations.
ND	(COMMON/SQLDS) - Value of subscript ℓ for which a distorted wake is used. If ND=NL=3: tip vortex $\vec{D}(\phi,\delta)$ is read in from cards punched by WG-71. If ND=0: Rigid wake case, no $\vec{D}(\phi,\delta)$ read in.
NH	n_h (COMMON/SAMB) - Number of azimuthal harmonics in harmonic analysis of downwash, airloads and blade motion.
NQ	When NG=NQ recompute influence coefficient matrix QMAT.
IS	(COMMON/SQLDS) - Shed wake indicator. IS=2: Always neglect far shed wake if IEPS=2: neglect near shed wake as well, if IEPS$\neq$2: include vortex sheet representation of near shed wake. IS$\neq$2: Include both near and far shed wake contributions.
IEPS	(COMMON/SQLDS) - near shed wake indicator IEPS=2: Concentrate near shed wake (wake within $\Delta\phi$ of η blade) in a vortex line $\Delta\phi$ behind η blade IEPS$\neq$2: Spread near shed wake over a vortex sheet from ϵ_s behind η blade to $\Delta\phi$ behind η blade.
IVLAC	(COMMON/BURST) - Controls use of lifting surface theory (LS) and tip vortex core bursting (VCB) during close blade-tip vortex interactions.

IVLAC	LS	VCB
2	no	yes
3	no	no
other	yes	yes

IABC (COMMON/SQLDS) - Indicates wake model used.

IABC	Tip Vortex	Inboard Trailing Wake	Shed Wake
1	VL	none	VL
2	VL	VL	VL
4	VS + VL	VL	VL
5	VS + VL	VS	VL

VL = vortex line, VS = vortex sheet

IVCM (COMMON/BURST) - Indicates vortex core model used

IVCM=1: Fixed wing type vortex core model,

$$\gamma(\rho) = \rho^2/(1 + \rho^2)$$

IVCM=2: Rotary wing vortex core model,

$$\gamma(\rho) = 0.2\rho^2/(1 + \rho^2), \quad \rho \leq 1.0$$

$$\gamma(\rho) = 1.0 - 0.9e^{-.075(\rho-1)}, \quad \rho \geq 1.0$$

ρ = (radial distance from vortex line)/(vortex core radius)

IRL (COMMON/SLDS) - Controls reading of ROTOR in subroutine LOADS and reading of XLDS in MAIN, program sets IRL=2 at end of each iteration. If (IRL≠2) read ROTOR, XLDS

IRB (COMMON/SLDS) - Controls reading of BEND in subroutine BLADE. If (IRB≠2) read BEND.

IPRF (COMMON/SLDS) - Read PERF in subroutine LOADS when NG>IPRF.

IOQM Number of radial stations for which QMAT is printed starting at blade tip, if IOQM=0 no print of QMAT

IPGM If IPGM=0 suppress print of Glauert Mach number correction

IPL	(COMMON/SLDS) - Controls printer plots of airloads by subroutine PLTL If IPL = 1 or NG $\geq$ IPL call PLTL
IPLX	(COMMON/PLT) - Controls reading, plotting, and harmonic analysis of experimental airloads (XLDS) IPLX=0: No XLDS read in. IPLX=2: Read, plot, and harmonic analyze XLDS
IUTQ	Controls use of tangential component of induced velocity (UTQ) in computation of tangential velocity (U_T) at point P_η. IUTQ=0: $U_T = \eta + \mu \sin(\psi)$ IUTQ=1: $U_T = \eta + \mu \sin(\psi) + UTQ$ IUTQ=2: $U_T = \left\{\begin{matrix} \eta + \mu \sin(\psi) \\ \eta + \mu \sin(\psi) + UTQ \end{matrix}\right\}$ (both cases done)
MEP	(COMMON/SBVGS) - Value of k for smallest η_k considered when searching for radial peak circulation to determine bound circulation at each azimuth ϕ.
NPOP	(COMMON/SQLDS) - Integer ratio $(\Delta\phi/\Delta\psi)$, $\Delta\phi = (NPOP)\Delta\psi$
NPRP	(COMMON/SQLDS) - Propagate tip vortex core bursting one additional $\Delta\phi$ step for every NPRP $\Delta\psi$ steps of blade motion since initial bursting.
NRNG	(COMMON/SQLDS) - Stop tip vortex core bursting when k_ϕ = KPH1 = NPRP $[\phi = (k_\phi - 1)\Delta\phi]$.
NDLM	(COMMON/SQLDS) - Maximum azimuthal extent of input tip vortex distortion, measured in $\Delta\psi$, $[(NDLM)\Delta\psi \leq 2\pi m]$.

E.2.3 NAMELIST/ROTOR

EPS	ε(COMMON/SLDS) - Rotor blade flapping hinge offset (nonD by R)

LN	LN(COMMON/SLDS) - Rotor blade Locke Number (ratio aerodynamic forces/inertial forces) $LN = \frac{2\pi\rho c R^4}{I_b}$, I_b = rotor blade flapwise moment of inertia about flapping hinge (slug-ft^2)
ALPS	α_s(COMMON/SLDS) - Stall angle of attack used in blade motion solution - dynamic stall used only in airloads computations given blade motion (input in degrees - used in radians).
CON	(COMMON/SLDS) - Dimensionalizing constant for airloads: $CON = \frac{\rho c (\Omega R)^2 \text{ (lb/ft)}}{2 \text{ (12 in/ft)}} = \text{(lb/in)}$.
F	F(COMMON/SLDS) - Theodorsen lift deficiency function used to approximate shed wake effects for zeroth iteration only.
EMT	M_T(COMMON/SLDS) - Rotor tip Mach number.
TC(6)	t_k(COMMON/SLDS) - Rotor blade thickness/chord ratio at radial stations η_k.
TW(6)	$\theta_k(\eta_k)$ (COMMON/SLDS) - Rotor blade twist at radial station η_k relative to η=0 (radians) $\theta(\eta_k) = \theta_o + \theta_k(\eta_k)$
IL	Controls type of blade motion solution: HSL = harmonic solution, BLADE = numerical solution IL = 0: HSL with no second harmonics IL = 1: Both HSL and BLADE airloads IL = 2: HSL airloads only IL = 3: BLADE airloads only (note: HSL blade motion used as initial guess in BLADE)
IPRG	Controls use of full or reduced airloads print (IPRG=0 or NG≥IPRG): full print; otherwise reduced print.
IPH	Controls harmonic analysis of airloads. (IPH=1 or NG≥IPH): harmonic analysis

IPUN Controls punch of angle of attack $\alpha(\eta,\psi)$.

(IPUN=0 or NG$\geq$IPUN): punch $\alpha(\eta,\psi)$

E.2.4 NAMELIST/BEND

Define: R = rotor radius (ft)

ε = flapping hinge offset (nonD by R)

$L = (1 - \varepsilon)R$ = length of blade outboard of hinge

$\bar{\varepsilon} = \varepsilon(R/L) = \dfrac{\varepsilon}{1 - \varepsilon}$

$\eta = r/R$ = radial coordinate (nonD by R)

$\bar{\eta} = \dfrac{\eta - \varepsilon}{1 - \varepsilon}$ radial coordinate from hinge to tip

$m(\bar{\eta})$ = radial blade mass distribution (slug/ft)

Ω = rotor angular velocity (radians/sec)

$a = \dfrac{\partial C_L}{\partial \alpha}$ = lift curve slope (per radian)

c = blade chord (ft)

ρ = air density (slug/ft^3)

I_b = blade flapwise moment of inertia about flapping hinge (slug-ft^2)

LN = Locke Number $2\pi\rho c R^4/I_b$

EI = flapwise stiffness (lb-ft^2)

The mass integrals are given in two forms:

a) the general form with $m(\bar{\eta})$

b) the special case $m(\bar{\eta}) = m$ constant, $I_b = \dfrac{mL^2}{3}$,

$$LN = \frac{2\pi\rho c R^4}{mL^3/3} = \frac{6\pi\rho c R}{m}\left(\frac{1}{1 - \varepsilon}\right)^3$$

M1

$$M_1 = \frac{2}{\rho a c R}\int_0^1 \bar{\eta}\,(\bar{\varepsilon} + \bar{\eta})\, m \, d\bar{\eta}$$

$$M_1 = \frac{2\pi}{LN\,a}\left(\frac{1}{1-\varepsilon}\right)^3 \left[2 + \left(\frac{3\varepsilon}{1-\varepsilon}\right)\right]$$

M2

$$M_2 = \frac{2}{\rho a c R} \int_0^1 \bar{\varepsilon}\left(4\bar{\eta}^2 - 3\bar{\eta}\right) m \, d\bar{\eta}$$

$$M_2 = \frac{-2\pi}{LNa} \left(\frac{1}{1-\varepsilon}\right)^3 \left(\frac{\varepsilon}{1-\varepsilon}\right)$$

M4

$$M_4 = \frac{2}{\rho a c R} \int_0^1 (\bar{\varepsilon} + \bar{\eta})\left(\frac{64}{3}\bar{\eta}^3 - 24\bar{\eta}^2 + 9\bar{\eta}\right) m \, d\bar{\eta}$$

$$M_4 = \frac{2\pi}{LNa} \left(\frac{1}{1-\varepsilon}\right)^3 \left[\frac{38}{5} + 11\left(\frac{\varepsilon}{1-\varepsilon}\right)\right]$$

M5

$$M_5 = \frac{2}{\rho a c R} \int_0^1 \bar{\eta}^2 m \, d\bar{\eta}$$

$$M_5 = \frac{4\pi}{LNa} \left(\frac{1}{1-\varepsilon}\right)^3$$

M6

$$M_6 = \frac{2}{\rho a c R} \int_0^1 \left(16\bar{\eta}^4 - 24\bar{\eta}^3 + 9\bar{\eta}^2\right) m \, d\bar{\eta}$$

$$M_6 = \frac{3}{5} M_5$$

M7

$$M_7 = \frac{128(1-\varepsilon)^{-4}}{\rho a c R \Omega^2 R^4} \int_0^1 E I \, d\bar{\eta}$$

If ω_2 = natural frequency of 1st mode flapwise bending

$$M_7 = M_6(\omega_2)^2 - M_4 - 9K$$

K

$$K = \frac{2k}{\rho a c R \Omega^2 R^3} \left(\frac{1}{1-\varepsilon}\right)^3$$

Where k(lb-ft) = spring constant of spring at flapping hinge used to simulate a cantilever rotor

NQIM Maximum number of blade motion solution iterations in BLADE

MQCM Maximum number of times $\Delta\phi$ can be halved in BLADE when attempting to obtain convergence of blade motion

NTIM	Maximum number of iterations on collective pitch in BLADE when attempting to match required C_T = CTO
EQ	Maximum difference between amplitude of blade motion (or its slope) at $\psi = 0$ and at $\psi = 360°$. If the difference is greater than EQ not converged, therefore iterate.
ECT	Maximum difference between actual C_T and required C_T = CTO. If difference > ECT not converged, therefore iterate.

E.3 Sample Input Data for LDS-73

& GENDAT

EMU = $\mu = 0.18$

AMBDA = $\lambda = \mu \tan i + \frac{C_T}{2\sqrt{\lambda^2+\mu^2}} = .0243$

EM = $m = 2$ - ($m \simeq 0.4/\mu$ about right)

RHOK = $\rho_k \doteq .05\,(c/R) = .0025$

DPSI = $\Delta\psi = 15^\circ$

AO = $a_0 = .085$ radians - (estimate eqn. 3.3)

BOER = $b/2R = .0122$

TANMU = $\mu \tan i \doteq \mu \left(\frac{f_q}{W}\right) = .009$ - (or eqn. 3.4)

GZER = $\gamma_0 = \frac{2\,C_T/\sigma}{\pi(1-\ell_2^2)} = .067$

EPSS = $\varepsilon_s = 0.5$ semi-chords - (Miller [9])

CTO = $C_T = \frac{T}{\rho\,\pi R^2(\Omega R)^2} = .0055$

SM = $\sigma = \frac{n_b c}{\pi R} = .0622$

A = $a = \frac{dC_L}{d\alpha} = 5.73$ per radian

THI = $\theta_1 = -.14$ radians (-8°)

RHOG = $\rho_g = .10$ - (typical)

WF = $W_f = 0.5$

VT = $V_T = \Omega R = 662$ ft/sec

DEL = 3.0

ELMN(6) = -9.,-9.,-3.,-3.,-3.,-3. (plotting parameters)

QMO = $q_{m_0} = .0020$ - ($q_{m_0} \simeq .1\lambda$ gives reasonable print)

ETA(6) = η_k = .25, .40, .55, .75, .85, .95

DETA(6) = $\Delta\eta_k$ = .225, .15, .225, .1, .1, .1

EL(3) = l_l = .10, .475, 1.0

DVB = 0.0001 -(do not make exactly 0.0)

FVB = 20.0001

ELI(25) = .8, 3*.7, 8*.475,6*.7,.8,.7,5*.8

ELT(25) = 6*.9,1.0,1.0,10*.9,1.0,1.0,5*.9

FTO(25) - 5*.75,.5,1.0,1.0,.75,9*.5,1.0,1.0,.75,.9,3*.75

FTD = .050

VSM = 0.20

RMVS $=\rho_{vs}$ = 0.0025 - (same as ρ_k)

DAM = 0.05 radians

Note: On choice of η, $\Delta\eta$, ℓ: see Table for a good example of choice of

$\Delta\eta$ give η

	ℓ_1				ℓ_2								ℓ_3
$\ell_1 + \Delta\eta$	.1		.325		.475		.70		.80		.90		1.0
η		.25		.40		.55		.75		.85		.95	
$\Delta\eta$		.225		.15		.225		.10		.10		.10	

Start at ℓ_1 and end at ℓ_3, try to center η between adjacent $\ell_1 + \Delta\eta$, take ℓ_2 as $\ell_1 + \Delta\eta$ value near 0.5.

Note: ELI, ELT, FTO are based on circulation distribution from previous runs.

& OPTOU

NB = n_b = 4 blades - (maximum of 6)

NE = n_η = 6 - (maximum of 6)

NL = n_ℓ = 3

NC = 1 case

NI = 5 - (iterations on circulation)

NG = 0 - (start with zeroth guess)

IG = 2 - (start with zeroth guess)

ND = 0 - (rigid wake)

NH = 6 - (harmonics)

NQ = 4 - (recompute QMAT when NG = 4)

IS = 1 - (include shed wake)

IEPS = 1 (use vortex sheet near shed wake)

IVLAC = 1 - (use both LS and VCB)

IABC = 4 - (VS + VL tip vortex)

IVCM = 1 - (fixed wing vortex core model)

$\left.\begin{array}{l}\text{IRL = 1}\\ \text{IRB = 1}\end{array}\right\}$ (Read ROTOR, BLADE)

IPRF = 9 > NI - (do not read PERF)

IOQM = 6 - (print all radial stations of QMAT)

IPGM = 1 - (print Glauert Mach correction)

IPL = 4 - (plot airloads when NG = 4)

IPLX = 1 - (plot XLDS)

IUTQ = 0 - (no induced contribution to U_T)

MEP = 3 - (only search $\eta \geq \eta_3 = .55$ for $\gamma(\eta)$ peak)

NPOP = 1 - ($\Delta\phi = \Delta\psi$)

NPRP = 3

NRNG = 16

NDLM = 49 - (not needed since ND = 0)

XLDS cards go here

& ROTOR

EPS = ε = .0357

LN = Locke number = 9.7

ALPS = α_s = 20°

CON = $\dfrac{\rho c(\Omega R)^2}{24}$ = 53.8 lb/in

F = 1.0 - (Theodorsen function not used)

EMT = M_T = 0.59

TC = 6*0.12 (12% thick blades)

TW = 0.035, -.056, -.077, -.105, -.119, -.133

(Note TW = $\theta_1\eta$ for linear twist)

IL = 3 - (BLADE airloads only)

IPRG = 4 - (full print when NG = 4)

IPM = 4 - (harmonic analysis when NG = 4)

IPUN = 4 - (On NG = 4 do punch $\alpha(\eta,\psi)$)

& BEND

M1 = $M_1 = \dfrac{2\pi}{LNa}\left(\dfrac{1}{1-\varepsilon}\right)^3\left[2+\left(\dfrac{\varepsilon}{1-\varepsilon}\right)\right] = .267$

M2 = $M_2 = \dfrac{-2\pi}{LNa}\left(\dfrac{1}{1-\varepsilon}\right)^3\left[\dfrac{\varepsilon}{1-\varepsilon}\right] = -.00451$

M4 = $M_4 = \dfrac{2\pi}{LNa}\left(\dfrac{1}{1-\varepsilon}\right)^3\left[\dfrac{38}{5}+11\left(\dfrac{\varepsilon}{1-\varepsilon}\right)\right] = 1.010$

M5 = $M_5 = \dfrac{4\pi}{LNa}\left(\dfrac{1}{1-\varepsilon}\right)^3 = .253$

M6 = $M_6 = \dfrac{3}{5}M_5 = .152$

M7 = $M_7 = \dfrac{128\,(1-\varepsilon)^{-4}}{\rho a c R \Omega^2 R^4}\,\overline{EI} = .097$

K = 0 - (articulated rotor)

NQIM = 25

NQCM = 1

NTIM = 5 (typical good values)

EQ = .00001

ECT = .00002

IPRQ = 4 - print when NG = 4

IPRQ Controls print of components of airloads.
If (IPRQ = 0 or NG≥IPRQ) print.

E.4 Input Variables Whose Values are Changed by the Execution of LDS-73

It if is desired to run several versions of the same basic case, with only minor variations, it is desirable to know which inputs must be reset because they have been changed by the execution of the basic case.

LDS-73

GENDAT - none (but GENDAT is always read)

OPTCOU - NG, IG, ND, IRL, IRB
NG = NI at end of a case
IG = 0 at end of each iteration
For distorted wake case, make ND = 3 On
first basic case thereafter make ND = 4
(reset ND = 3 by program) to avoid reading
$\vec{D}$ again.
IRL = IRB = 2 at end of each iteration. They
only need to be reset if it is necessary to
change something in ROTOR or BLADE

E.5 LDS-73 Outputs

E.5.1 Printed Output

The printed output is very extensive to provide a detailed picture of the source of the airloads and to aid debugging. For most cases, all of this output is not needed and various indicators are included in the input data which can be used to suppress portions of the output. The output will be described here for the case where almost nothing is suppressed. It is presented in the order it would be printed out.

1) Label, Wake Model, CASE=IABC - output by MAIN

2) Input data from GENDAT and OPTCOU - output by PNTIN

3) If (IPGM.NE.0) Glauert Mach number correction for lift curve slope GMC(η,ψ) - output by MAIN

4) Iteration number - output by MAIN

5) If (IG.LE.1) Bound circulation data - output by GAMT: ELB = ℓ_2, GBVR = $\gamma(\eta,\psi)$
   GBVS not used, GBV(3) = $\gamma_{bT}(\phi)$, GS(3) = $\gamma_s(\phi)$
   GLMX(ψ) = $\gamma(\eta_{NE},\psi)/\gamma_{MAX}(\psi)$, GTMX
   is GLMX extrapolated from η_{NE} to $\eta = 1.0$

6) If (IG.LE.1) Print plot of $\gamma_{bv}(\phi)/\gamma_o$ for both the last (*) and next to last (+) iterations. If both iterations plot at the same point, only the (*) is printed for that point.

7) Special print for any vortex line or sheet segment whose contribution to the induced velocity at point P_η (QL) is greater than QMD.

   The vortex sheet print is one line and comes from QVS (far wake) or QLDS (near wake). The vortex line print is two lines long and comes from subroutine QSVL. There is another one line print from QLDS whenever vortex bursting first occurs. This can be distinguished from the vortex sheet line by the first symbols. The bursting line starts (IJK=) while the vortex sheet line starts (I=). The three different types of printout are mixed up randomly as they occur in the computation. For details on these printouts see the listings of the appropriate subroutines in Appendix F.

8) If (IOQM).GT.0) QMAT (η,ψ,ϕ) - output by MAIN

9) Tangential induced velocity component $q_t(\eta,\psi)$ and tangential velocity $U_T = \eta + \mu \sin\psi$ - output by MAIN

10) Downwash perpendicular to tip path plane TPP

$\lambda(\eta,\psi) = \mu \tan i + \sum_{\phi} \gamma_{bv}(\phi)$*QMAT$(\eta,\psi,\phi)$ - output by MAIN

11) Downwash contributions of individual blades (ZETA = ζ) and of tip vortex ($\ell = 1.0$), inboard trailing wake ($\ell = \ell_2$), and shed wake ($\ell = \ell_1$), where EL = ℓ.

SUM(η,ψ,ζ,ℓ) = SUM(K,J,LB) where K is η subscript, J is ψ subscript, and LB is a combined ζ,ℓ subscript - output by MAIN

12) $\lambda(\eta,\psi) = \mu \tan i + \sum_{\ell}\sum_{\zeta}$ SUM(η,ψ,ζ,ℓ), should be the same as item (10), and provides a check on QMAT - output by MAIN

13) Harmonic analysis of $\lambda(\eta,\psi)$ - output by MAIN

14) Input data from ROTOR - output by LOADS

15) Harmonics of rigid body flapping and the contributions of thrust (C_T), twist (θ_1), downwash harmonics $(\lambda_o, \lambda_{2c}, \lambda_{1c}, \lambda_{2s})$, and second harmonic flapping (b_2, a_2) to the harmonic blade motion solution. It is assumed: $\beta(\psi) = a_o - a_1 \cos\psi - b_1 \sin\psi - a_2 \cos 2\psi - b_2 \sin 2\psi$. The radial integrals of various λ harmonics times η or $\frac{\eta - \varepsilon}{1 - \varepsilon}$ are also printed - output by HSL.

16) If (IL.LT.3) Airloads are computed using HSL blade motion data. These can be output here, the same way the blade bending airloads are output by PNTLDS, as described below.

17) Input data from BEND - output by BLADE

18) Harmonics of blade motion Q1 = $q_1(\psi) = \beta(\psi)$
Q2 = $q_2(\psi)$ = first mode blade bending, harmonic analysis
is with + signs unlike $\beta(\psi)$ in item (15) - output by BLADE

19) Blade motion Q1 = $q_1(\psi)$, QD1 = $dq_1/d\psi$, Q2 = $q_2\ (\psi)$, QD2 = $dq_2/d\psi$, the $\cos\psi$ and $\sin\psi$ are just a check on the accuracy of these tables - output by BLADE

20) If (IPRQ.LE.O.OR.NG.GE.IPRQ) Components of airloads: LOADS = $L^*(\eta,\psi)$, where airloads (lb/in)

$L(\eta,\psi) = \frac{\rho c V_T^2}{24}\ \frac{dC_L}{d\alpha}\ L^*\ (\eta,\psi)$, ALPHA = $\alpha(\eta,\psi)$,

THETA = $\theta(\eta)$, UP/UT = $U_p(\eta,\psi)/U_T(\eta,\psi)$,

UT = $U_T(\eta,\psi)$ = tangential velocity at point P_η,

UP = $U_p(\eta,\psi)$ = perpendicular velocity at point P_η,

LAMBDA = $\lambda(\eta,\psi)$, MU*A1 = μa_1 - converts

$\lambda(\eta,\psi)$ from TPP to NFP, Q1 = $q_1(\psi)\ \mu \cos\psi$,

Q2 = $q_2(\psi)\ \left(\frac{d\gamma_2}{d\eta}\right)\mu\cos\psi$, QD1 = $\eta\frac{dq_1}{d\psi}$,

QD2 = $\gamma_2(\eta)\left(\frac{dq_2}{d\psi}\right)$- where $\gamma_2(\eta)$ = first blade bending mode shape,

DZDR = $\frac{dz(\eta,\psi)}{d\eta}$ where $z(\eta,\psi)$ is displacement of point P_η perpendicular to TPP.

Note: UP=LAMBDA+MU*A1+Q1+Q2+QD1+QD2, ALPHA=THETA - $\tan^{-1}$(UP/UT) - output by BLADE

21) Airloads (lb/in) $L(\eta,\psi)$ and angle of attack (degrees) $\alpha(\eta,\psi)$ - output by PNTLDS

22) If (IPRG.LE.O.OR.NG.GE.IPRG) Static stall angle of attack (degrees) $\alpha_s(\eta,\psi)$ and reference angle of attack (degrees) $\alpha_R(\eta,\psi)$ - which is used in dynamic stall model - output by PNTLDS

23) If (IPRG.LE.O.OR.NG.GE.IPRG) Thrust perpendicular TPP (lb), Azimuthal Force in TPP (lb), Radial Force in TPP (lb), Horizontal Force in TPP (lb) output by PNTLDS

24) If (IPRG.LE.O.OR.NG.GE.IPRG) Section drag coefficient C_D, CDM increment to C_D due to Mach number, blade element Mach number $M(\eta,\psi)$, drag divergence Mach number $M_D(\eta,\psi)$ - output by PNTLDS

25) If (IPRG.LE.O.OR.NG.GE.IPRG) Airloads (lb/in) $L(\eta,\psi)$ - same as in (21), airloads using old simple dynamic stall model - still used in blade motion computations, angle of attack if $dC_L/d\alpha$ were not modified by dynamic stall, actual angle of attack - output by PNTLDS

26) If (IPRG.LE.O.OR.NG.GE.IPRG) Angle of attack $\alpha(\eta,\psi)$ - same as in (21) and (25), Blade element contribution to $C_T \times 10^6$ - output by PNTLDS

27) Blade element contribution to parasite torque coefficient $C_{Q_P} \times 10^6$, blade element contribution to induced torque coefficient $C_{Q_I} \times 10^6$ - output by PNTLDS

28) C_T(thrust), C_H(horizontal force), C_x(propulsive force), C_Q(torque), C_{Q_P} (parasite torque), C_{Q_I} (induced torque), L/D_E of aircraft, L/D_E of rotor alone, $C_L^* = \frac{L}{qd^2\sigma}$, C_x^* from $\frac{X}{qd^2\sigma}$ - output by LDS

29) If (IPH.EQ.1.OR.IPH.LE.NG) Harmonic analysis of airloads $L(\eta,\psi)$ - output by HALDS

30) If (IPL.EQ.1.OR.IPL.LE.NG) Printer plots of airloads $L(\eta,\psi)$ - output by PLTL

31) Iteration number and execution times - output by MAIN

32) Bound circulation - same as (5) except that GBVR is $\gamma(\eta,\psi)$ from the most recent iteration while GBVS is $\gamma(\eta,\psi)$ from the next most recent iteration. This allows a convergence check - output by GAMT

33) Same as item (6)

34) Same as items (9) and (10)

35) Same as items (13) through (30)

36) For each iteration repeat items (31) through (35)

37) On last iteration only - repeat items (1) and (2) and then output vortex bursting ages KPB(ψ) - output by MAIN

38) Same as items (31) through (33)

The SUM(η,ψ,ζ,ℓ) array cannot be computed from the QMAT(η,ψ,ϕ) array because QMAT has already been summed over ℓ and ζ to save storage. Therefore, items (7), (11), and (12) cannot be repeated after the first iteration unless the induced velocity computation is repeated, which is very expensive.

To save printing costs, items (20), (22) through (26), (29), and (30) are normally suppressed for all but the last iteration. Item (16) is not normally used because airloads are not normally computed using HSL blade motion. The airloads plots (30) and items (37) and (38) are intended to be separated from the rest of the output to provide a summary of the output for one case. The bulky main output need only be referred to when extra detail is required.

In the event of an overflow, underflow, divide check, or square root of a negative argument, the extended error handling facility is used to call subroutine FXSQRT. This subroutine prints a large dump of every variable in every COMMON statement as an aid to debugging. The unformated, namelist output is used. If the various iterations in subroutine BLADE do not converge within a specified number of tries, then some special output is printed by BLADE.

E.5.2 Punched Output

The only punched output is the angle-of-attack distribution $\alpha(\eta,\psi)$. This is normally the only output on the last iteration. It can be used to provide a basis for the computation of the bound circulation $\gamma(\eta,\psi)$ in a WG-71 run or in a further LDS-73 run.

If (IPUN.EQ.O.OR.NG.GE.IPUN) Punch $\alpha(\eta,\psi),\eta$

```
          DO 10 K = 1, NE
          PUNCH 20, ETG(K)
    10    PUNCH 30, (ALPH(K,J),J=1,NPSI)
    20    FORMAT (F6.3)
    30    FORMAT (5E14.7)
```

The punched out η are only used to help visual identification of the $\alpha(\eta,\psi)$. The η cards must be removed before the $\alpha(\eta,\psi)$ cards are input to either LDS-73 or WG-71.

E.6 LDS-73 COMMON Statements

E.6.1 COMMON/SQVLS/

Used in: MAIN, QLDS, QSVL, QVS, FXSQRT

I	Index $i = \frac{n_b \zeta_i}{2\pi} + 1$, ($i = 1, n_b$), indicates which blade on an n_b bladed rotor is the ℓ blade
J	Index $j = \frac{\psi_j}{\Delta\psi} + 1$, ($j = 1, n_\psi$), indicates azimuth angle ψ_j of η blade
K	Index k for η_k ($k = 1, n$), indicates radial station η_k at which induced velocity is computed
KP	Index $k_\delta = \frac{\delta_{k_\delta}}{\Delta\psi} + 1$, ($k_\delta = 1, n_\delta$), indicates age δ_{k_δ} of wake element shed by ℓ blade
KPHI	Index $k_\phi = \frac{\phi_{k_\phi}}{\Delta\psi} + 1$, ($k_\phi = 1, n_\phi$) indicates azimuth angle ϕ_{k_ϕ} of ℓ blade, although ϕ_{k_ϕ} goes up to $2\pi m$ and $\Delta\phi$ may be $\neq \Delta\psi$, k_ϕ is based on $\Delta\psi$ and is cyclic (i.e., if $\phi' = \phi'' + 2\pi$ then $k_\phi' = k_\phi''$)
L	Index ℓ for ℓ_ℓ ($\ell = 1, n_\ell$) indicates radial station on ℓ blade from which wake element is shed
IZ	Indicates type of induced velocity computation in subroutines QVS and QSVL - see subroutine listings.
XA,YA, ZA,XB, YB,ZB	$x_a, y_a, z_a, x_b, y_b, z_b$ - The xy,z components of $\vec{a}$ and $\vec{b}$ extending from point P_η on the η blade to endpoints P_a and P_b of the vortex line segment (or of the front edge of the vortex sheet segment), (nonD by R)
ZM	Z_m - In vortex sheet computations the mean z component of vectors from point P_η to the endpoints of the back edge of the vortex sheet (nonD by R)

VSM	Vortex sheet margin (Input - GENDAT) - If necessary the length of a vortex sheet segment is extended so that the projection of point P_η onto the vortex sheet is never closer than VSM times the length $(\Delta\tilde{x})$ of the sheet to either the front or back edges (those edges parallel to the circulation vector).
DELL	$\Delta\tilde{x}$ - Length of a vortex sheet segment perpendicular to its circulation vector (nonD by R)
GA	$\gamma_a \frac{b}{2R}$ - NonD (by $2\pi b\Omega R$) circulation of vortex line (or vortex sheet) segment times rotor blade semichord over twice the rotor radius
QT	$\frac{q_t}{(\gamma_a\ b/2R)}$ - Tangential component of the induced velocity contribution of a vortex line (or sheet) segment at point P_η divided by GA times tip speed
QK	$\frac{q_k}{(\gamma_a\ b/2R)}$ - Downwash (k component of induced velocity) contribution of a vortex line (or sheet) segment at point P_η divided by GA times tip speed
RHVS	ρ_{vs} (Input - GENDAT) Vortex sheet core radius (nonD by R)
DAM	$\Delta\alpha_m$ (Input - GENDAT) Maximum change in local angle of attack due to any one vortex line or vortex sheet segment (radians)
QMO	q_{mo} (Input - GENDAT) - Minimum value of the induced velocity contribution of any vortex line or vortex sheet segment for which a special printout is made in QSVL or QVS (nonD by tip speed)

E.6.2 COMMON/BURST/

Used in: MAIN, PNTIN, QLDS, QSVL, FXSQRT

IVLAC	(Input - OPTCOU) - Controls use of lifting surface theory (LS) and of vortex core bursting (VCB) during close blade-tip vortex interactions

IVLAC	LS	VCB
2	no	yes
3	no	no
other	yes	yes

IVCM (Input - OPTCOU) - Indicates vortex core model used.

IVCM=1: Fixed wing-type vortex core model, $\gamma(\rho) = \rho^2/(1 + \rho^2)$

IVCM=2: Rotary wing-type vortex core model

$$\gamma(\rho) = \begin{cases} 0.2\,\rho^2/(1+\rho^2) & , \rho \leq 1.0 \\ 1.0 - 0.9\, e^{-.075(\rho-1)} & , \rho \geq 1.0 \end{cases}$$

DVB (Input - GENDAT) If $\Delta\ell_{xy}$ < DVB then delay tip vortex core bursting by $[1 + 2(\text{DVB} - \Delta\ell_{xy})]$ (truncated to an integer) $\Delta\phi$ intervals. This controls how much of any tip vortex line segment must have passed close to the η blade before the entire segment is burst.

FVB (Input - GENDAT) If a tip vortex passes within $(\text{FVB})\rho_c$ above or below the η blade and the nearest end of the vortex line segment is between one segment length behind η blade and one-half segment length ahead of η blade a close blade-tip vortex interaction has occurred. (Default values DVB=0.001, FVB=2.001)

ETAC η_c - Radial location of the passage of a vortex line segment or its extension above or below η blade or its extension (nonD by R)

DLXY $\Delta\ell_{xy}$ - Distance in TPP (Tip Path Plane) from nearest end of a vortex line segment to the η blade or its extension, measured parallel to the segment (nonD by segment length)

ZC z_c - Z component of the distance from a vortex line segment or its extension to the η blade at the point of passage above or below the η blade (nonD by R)

RHOC	ρ_c - Vortex core radius = radial distance from the center to the maximum induced velocity. Used in QSVL (nonD by R)
RHOG	ρ_g (Input - GENDAT) Burst vortex core radius (nonD by R)
ABOER	$a\ (\frac{b}{2R})$ (Inputs - GENDAT) $a = \frac{\partial C_L}{\partial \alpha}$, b = rotor blade semichord, R = rotor radius
ETA	η - Radial station on η blade of point P_η at which induced velocity is computed (nonD by R)
AO	a_o (Input GENDAT) - Coning angle (zeroth harmonic of rigid blade flapping). This input value of a_o is used only for determining $\vec{a}$ and $\vec{b}$ from P_η to P_a and P_b. The value of a_o computed in the blade motion portion of LDS-73 is a separate variable (radians)
HRS	$\rho^2 = (h/\rho_c)^2$, h = Radial distance from the center of the vortex line to point P_η (nonD by R) - ρ_c = vortex core radius (nonD by R)
GK	$\gamma(\rho)$ - Radial circulation distribution of a vortex line normalized so that $\gamma(\infty) = 1.0$ and $\gamma(1.0)$ gives the maximum induced velocity. Fixed wing vortex core model: $\gamma(\rho) = \rho^2/(1+\rho^2)$ Rotary wing vortex core model: $\gamma(\rho) = 0.2\rho^2/(1+\rho^2) \quad , \quad \rho \leq 1.0$ $\gamma(\rho) = 1.0 - 0.9\, e^{-.075(\rho-1)} \quad , \quad \rho \geq 1.0$

E.6.3 COMMON/SINCO/

Used in: MAIN, GAMT,QLDS,QSVL,QVS,LOADS,BLADE,FXSQRT

S(97)	$\sin(\psi)$ tabulated every $\Delta\psi$ from $\psi = 0°$ to $\psi = 360°$
C(97)	$\cos(\psi)$ tabulated every $\Delta\psi$ from $\psi = 0°$ to $\psi = 360°$
UTS(6,25)	$U_T(\eta,\psi)$ = local tangential velocity parallel to blade chord $U_T(\eta,\psi) = \eta + \mu \sin\psi + q_t(\eta,\psi)$ where $q_t(\eta,\psi)$

is the tangential component of the induced velocity (nonD by tip speed)

E.6.4 COMMON/SGAM/

Used in: MAIN, PNTIN,GAMT,QLDS,LOADS,MSL,BLADE,HALDS,HART,PNTLDS, PLTL,FXSQRT

NE	n_η(Input OPTCOU - number of radial stations η_k on η blade (NETA in: LOADS,HSL,BLADE,HALDS,PNTLDS)
NPS1	$n_\psi = \frac{2\pi}{\Delta\psi} + 1$ - Number of azimuth angles ψ considered. (NPS10 in BLADE)
NPS	NPS1-1 (NPS0 in BLADE)
NG	(Input OPTCOU) - Current iteration number, increased by one after each iteration
WF	w_f (Input GENDAT) - Weighting factor, to start a new iteration for circulation or blade motion, take w_f times latest results plus $(1 - w_f)$ times previous results
EMU	μ(Input GENDAT) - Advance ratio (V/ R)
A	a (Input GENDAT) - $\frac{\partial C_L}{\partial \alpha}$. This input value remains constant during execution and various factors are used in blade motion and airloads computations to modify the effective $\frac{\partial C_L}{\partial \alpha}$ (per radian)
BOER	$\frac{b}{2R}$ (Input GENDAT) - b = rotor blade semichord, R = rotor radius
DANG	$\Delta\psi$ - Azimuth angle interval size (radians), (DANGO in BLADE)
SUM(6, 25,18)	$q_z(\eta,\psi,\zeta,\ell)$ - z component of the contribution to the induced velocity at point P_η of the wake from radial station ℓ on the ζ blade. There is provision for 6η, 25ψ, and 18ζ, ℓ combinations (6ζ and 3ℓ) (nonD by tip speed)

Note: After SUM(η,ψ,ζ,ℓ) is printed out, other arrays are stored in the same space by means of EQUIVALENCE statements

ETG(6) η_k (Input GENDAT) - Radial stations on η blade of point P_η at which airloads are computed (nonD by R), (ETP(6) in: GAMT, BLADE)

DETA(6) $\Delta\eta_k$ (Input GENDAT) - Delta η_k used in radial integrations (nonD by R), (DETP(6) in GAMT)

ANG(25) ψ_j - Azimuth angles of η blade measured in degrees, used only for output (U(25) in PLTL)

PSI(25) ψ_j - Azimuth angles of η blade measured in radians, used for computations.

ALPHA(6,25) $\alpha(\eta,\psi)$ - Angle-of-attack distribution as a function of radius (η) and azimuth (ψ) used for computation of circulation distribution, can be input or obtained from a previous iteration. (ALPH(6,25) in: LOADS,HSL,BLADE,HALDS,PNTLDS)

E.6.5 COMMON/SBVGS/

Used in: MAIN, PNTIN,GAMT,QLDS,FXSQRT

IG (Input OPTCOU) - Indicates type of circulation computation used.

IG	Circulation $[\gamma_{bv}(\psi)]$ computation used
0	Compute γ_{bv} using $\alpha(\eta,\psi)$ from previous iteration stored in ALPHA(6,25)
1	Compute γ_{bv} using $\gamma(\eta,\psi)$ read in from cards along with associated η_k and $\Delta\eta_k$ which can be different from η_k and $\Delta\eta_k$ in the remainder of the program
2	Zeroth iteration, set all $\gamma_{bv}(\psi) = \gamma_o$ (GZER - Input GENDAT) for all ψ, therefore γ_s should be zero, instead make γ_s small but finite so that it will not contribute to $\lambda(\eta,\psi)$ but its influence coefficient will be computed.

MEP (Input OPTCOU) - Value of k for smallest η_k considered when searching for radial peak circulation to determine bound circulation at each azimuth ϕ.

GZER γ_o (Input GENDAT) - Constant value of bound circulation used in zeroth iteration (IG>1) (nonD by $2\pi b\Omega R$)

FL(3) ℓ_ℓ (Input GENDAT) - Radial stations on ℓ blade of points P_ℓ from which elements of vortex wake are shed. ℓ_1 = inboard edge of inboard trailing vortex sheet, ℓ_2 = outboard edge of inboard trailing sheet or location of inboard trailing vortex line, ℓ_3 = location of tip vortex (normally $\ell_3 = 1.0$) (nonD by R)

BV(25) $\frac{b}{2R}\gamma_{bv}(\psi_j)$ - Bound circulation as a function of azimuth times $(\frac{b}{2R})$ (nonD by $2\pi b\Omega R$)

GS(25) $\frac{b}{2R}\gamma_s(\psi_j)$ - Shed circulation as a function of azimuth times $(\frac{b}{2R})$ (nonD by $2\pi b\Omega R$)

E.6.6 COMMON/SQLDS/

Used in MAIN, PNTIN, QLDS, FXSQRT

IABC (Input OPTCOU) - Indicates wake model used.

IABC	Tip Vortex	Inboard Trailing Wake	Shed Wake
1	VL	none	VL
2	VL	VL	VL
4	VL+VS	VL	VL
5	VL+VS	VS	VL

IEPS (Input OPTCOU) - If IEPS≠2: Spread near shed wake (from $\Delta\phi$ nearest the η blade) over a vortex sheet starting ε_s behind the blade. If IEPS=2: concentrate near shed wake (from $\Delta\phi$ nearest the η blade) in a vortex line $\Delta\phi$ behind the η blade.

IS (Input OPTCOU) - If IS≠2:Compute both near and far shed wake contributions to induced velocity.
If IS=2: Neglect far shed wake always, and if IEPS=2, neglect near shed wake as well, if IEPS≠2 include the vortex sheet representation of the near shed wake.

KZAD	$(\frac{2\pi}{n_b \Delta\psi})$ - Increment in ζ_i for each unit increment in i
NB	n_b (Input OPTCOU) - Number of rotor blades
NL	n_ℓ (Input OPTCOU) - Number of radial stations ℓ_ℓ on ℓ blade. For LDS-73, $n_\ell = 3$
ND	(Input OPTCOU) - Value of ℓ for which a distorted wake is used, $\vec{D}(\phi,\delta)$ is read in from cards output by WG-71, If ND=0: Rigid wake case, no $\vec{D}(\phi,\delta)$ read in.
NDEL	$n_\delta = (\frac{2\pi m}{\Delta\psi} + 1)$ - Maximum azimuth extent of wake, measured in $\Delta\psi$.
NPOP	(Input OPTCOU) - Integer ratio $\Delta\phi/\Delta\psi$
NPRP	(Input OPTCOU) - Propagate tip vortex core bursting one additional $\Delta\phi$ step for every NPRP $\Delta\psi$ steps of blade motion since initial bursting.
NRNG	(Input OPTCOU) - Stop tip vortex core bursting propagation at KPH1=NRNG
NDLM	(Input OPTCOU) - Maximum azimuthal extent of input tip vortex distortion, measured in $\Delta\psi$, ($NDLM \leq NDEL$)
FTD	f_t^D (Input GENDAT) - Tip vortex rollup rate, tip vortex line circulation as a fraction of total tip vortex circulation $f_t(\phi,\delta)$ is increased by $\Delta\phi * f_t^D$ every $\Delta\phi$ step in age until all tip vortex circulation is concentrated in the vortex line ($f_t(\phi,\delta) = 1.0$)
RHOK	ρ_k (Input GENDAT) - Nonburst value of ρ_c - vortex core radius (nonD by R)
DPHR	$\Delta\phi$ - Azimuth angle increment between successive positions of ℓ blade (azimuthal extent of trailing vortex line segment) (radians)
TANMU	$\mu\tan(i)$ (Input GENDAT) - Inflow (downwash) perpendicular to TPP due to flight speed (relative wind) (nonD by tip speed)

AMBDA	λ(Input GENDAT) - Uniform, average downwash perpendicular to TPP - used to determine rigid wake geometry (nonD by tip speed)
ZETA(6)	ζ_i - Azimuth angle difference between ith ℓ blade and η blade. For n_b bladed rotor: $\zeta_i = (i - 1)\frac{2\pi}{n_b}$
KPB(25)	$k_\delta^b(\phi)$ - Tip vortex core bursting occurs at azimuth ϕ when point P_ℓ reaches age $\delta \geq \Delta\psi[k_\delta^b(\phi)]$, $k_\delta^b(\phi) = 1 + (\delta_b(\psi)/(\Delta\psi)$
ELT(25)	$\ell_t(\phi)$ (Input GENDAT) - Radial station of inboard edge of tip vortex sheet at azimuth angle ϕ (nonD by R)
ELI(25)	$\ell_i(\phi)$ (Input GENDAT) - Radial station of outboard edge of inboard trailing vortex sheet at azimuth angle ϕ (nonD by R)
FTO(25)	$f_t^o(\phi)$ (Input GENDAT) - Fraction of total tip vortex circulation concentrated into tip vortex line at azimuth angle ϕ and zero age ($\delta = 0$)

E.6.7 COMMON/SAMB/

Used in: MAIN, LOADS,HSL,BLADE,HALDS,HART,FXSQRT

NH	n_h (Input OPTCOU) - Number of azimuthal harmonics in harmonic analysis of downwash airloads and blade motion
AMB(25, 6)	$\lambda(\psi,\eta)$ - Computed downwash perpendicular to TPP at point P_η - at azimuth ψ and radial station η (nonD by tip speed), (AMBDA(25,6) in: LOADS, HSL,BLADE,HALDS)
AMBO(6)	$\lambda_o(\eta)$ - Zeroth azimuthal harmonic of $\lambda(\psi,\eta)$
CLAM(9, 6)	$\lambda_{nc}(\eta)$ - nth cosine azimuthal harmonic of $\lambda(\psi,\eta)$
SLAM(9, 6)	$\lambda_{ns}(\eta)$ - nth sine azimuthal harmonic of $\lambda(\psi,\eta)$

E.6.8 COMMON/SLDS/

Used in: MAIN, PNTIN,LOADS,HSL,BLADE,PLTL,FXSQRT

IRB	(Input OPTCOU) - Controls reading of namelist BEND in subroutine BLADE. If IRB<2: read BEND, otherwise do not.
IRL	(Input OPTCOU) - Controls reading of namelist ROTOR in subroutine LOADS. If IRL≠2: read ROTOR. If IRL=2: do not read ROTOR
IPL	(Input OPTCOU) - Controls printer plots of airloads by subroutine PLTL. If IPL=1 or IPL≤NG (iteration number): call PLTL, otherwise do not.
IPRF	(Input OPTCOU) - Controls reading of namelist PERF in subroutine LOADS. If IPRF<NG (iteration number): read PERF. Otherwise do not.
IUT	Controls operations in subroutine LOADS when IUTQ=2; i.e., airloads are computed both with (IUT=9) and without (IUT=2) tangential induced velocity. If IUT=2: suppress punch out $\alpha(\eta,\psi)$, harmonic analysis of airloads and plots of airloads. If IUT=9: suppress reading of namelist ROTOR, printing airloads inputs and harmonic blade motion solution.
NI	(Input OPTCOU) - Number of circulation iterations to be run. When NG=NI, stop.
CTO	C_T (Input GENDAT) - Input target value of thrust coefficient, iteration on collective pitch (θ_o) until C_T is achieved. $C_T = \frac{\text{thrust(lb)}}{\rho\pi R^2(\Omega R)^2}$, ρ = air density (slug/ft^3), R = rotor radius (ft), ΩR = tip speed (ft/sec)
SM	σ(Input GENDAT) - Rotor solidity: $\sigma = \frac{\text{rotor blade area}}{\text{rotor disk area}} = \frac{cn_b}{\pi R}$ c = rotor blade chord (ft)
TH1	θ_1 (Input GENDAT) - Rotor blade linear twist (radians), $\theta(\eta) = \theta_o + \theta_1\eta$

VT	$V_T = \Omega R$ (Input GENDAT) - Rotor tip speed (ft/sec)
EPS	ε (Input ROTOR) - Rotor blade flapping hinge offset (nonD by R)
LN	LN (Input ROTOR) - Rotor blade Locke number (ratio aerodynamic forces/inertia forces) $LN = (2\pi\rho cR^4)/I_b$. I_b = rotor blade flapwise moment of inertia about flapping hinge (slug-ft^2).
F	F(Input ROTOR) - Theodorsen lift deficiency function used to approximate shed wake effects only for zeroth iteration or when IS=2 (no shed wake used)
CON	(Input ROTOR) - Constant used to obtain dimensional airloads: $CON = \frac{\rho c(\Omega R)^2 (lb/ft)}{2(12\ in/ft)}$
ALPS	α_s (Input ROTOR) - Stall angle of attack, read in as degrees used in radians
COSI	Cosine of angle i between relative wind and TPP
GMA	Average Glauert Mach number correction to $\frac{\partial c_L}{\partial \alpha}$; $\frac{\partial c_L}{\partial \alpha} = \frac{a}{GMA}$, $GMA = 1 - (.75M_T)^2$, M_T = rotor tip Mach number
ELO	EL(1)=ℓ_1 - Inboardmost radial station ℓ - normally radius of rotor blade root cutout (nonD by R)
OMEPS	1-EPS=$1-\varepsilon$ - Length of rotor blade outboard of flapping hinge (nonD by R)
EMT	M_T(Input ROTOR) - Rotor tip Mach number $(\frac{\text{tip speed}}{\text{speed of sound}})$
EMS	EMU*EMU=μ^2 - Advance ratio squared
DENTH	Denominator of expression for collective pitch (θ_o) - a very complicated function of μ, ε, and ℓ_1

AlMU	EMU*A1 = $\mu a_1 = (\lambda_{NFP} - \lambda_{TPP})$ - Inflow through NFP due to 1st harmonic flapping
AZERO	a_o - Zeroth harmonic of rigid blade flapping motion (coning angle). Computed value from blade motion solution not to be confused with AO which is input (radians)
AONE, BONE, ATWO, BTWO	a_1, b_1 a_2, b_2 - Harmonics of rigid blade flapping motion - $\beta(\psi) = a_o - a, \cos\psi - b, \sin\psi - a_2 \cos 2\psi - b_2 \sin 2\psi$ (radians)
THETAO	θ_o - Collective pitch at zero radius (radians)
ETE(6)	$\bar{\eta}_k = \dfrac{\eta_k - \varepsilon}{1 - \varepsilon}$ - Radial stations on blade measured from flapping hinge instead of center of rotation (nonD by L = $(1 - \varepsilon)R$), (ETA(6) in: LOADS, HSL,BLADE)
TC(6)	t_k (Input ROTOR) - Rotor blade thickness/chord ratio at radial stations η_k on η blade
ELDS(6)	$\sum_{\psi} L(\eta,\psi)$ - Sum over azimuth of airloads at radial station η - used to find zeroth harmonic of airloads
TW(6)	$\theta_k(\eta_k)$ (Input ROTOR) - Rotor blade twist at radial station η_k relative to $\eta = 0$ (radians) $\theta(\eta_k) = \theta_o + \theta_k(\eta_k)$
GMC(6, 25)	$G_M(\eta,\psi)$ - Glauert Mach number correction: $\dfrac{dc_L}{d\alpha} = a/G_M(\eta,\psi)$ $G_M(\eta,\psi) = \text{Max.} \left\{ \begin{array}{l} \sqrt{1 - (\eta + \mu \sin\psi)^2 M_T^2} \\ 0.60 \end{array} \right.$

E.6.9 COMMON/PLT/

Used in: MAIN, HALDS, PLTL

IPLX	(Input OPTCOU) - Controls reading in plotting and harmonic analysis of experimental airloads. IPLX 0 no experimental airloads IPLX 1 read in and plot exp. airloads IPLX 2 read in, plot and harmonic analyze experimental airloads
DEL	(Input GENDAT) - Interval size on airloads plots ordinate (lb/in)
ELMN(6)	(Input GENDAT) - Minimum value on airloads plot ordinate for each different η · (lb/in)
XLDS(6, 25)	(Read in if IPLX≠0) - Experimental airloads at radial station η and azimuth angle ψ for 6η and 25ψ (lb/in)

E.6.10 COMMON/SFX/

Used in: MAIN, QLDS, FXSQRT

PZAP	$\delta = \psi + \zeta - \phi$, Age of point P_η
GPS	$\frac{\varepsilon_s b}{R}$ - Distance behind η blade to start of near shed wake vortex (nonD by R)
QTAN(6, 25,25)	$\frac{q_t(\eta,\psi,\phi)}{\gamma_{bv}(\phi)}$ - Influence coefficient matrix which when multiplied by bound circulation $\gamma_{bv}(\phi)$ at azimuth angle ϕ gives tangential induced velocity at point P_η (radial station η and azimuth angle ψ) due to all wake elements involving bound circulation $\gamma_{bv}(\phi)$; i.e., all trailing wake elements having azimuth angle ϕ plus shed wake elements having azimuth angles ϕ and $\phi - \Delta\phi$ (nonD by tip speed)
VI(3),VK(3), VK(3)	$\vec{i}, \vec{j}, \vec{k}$ (x,y,z) components of $\vec{a}$ and $\vec{b}$ from point P_η on η blade to points P_a and P_b in wake as a function of ℓ - radial station on ℓ blade at azimuth angle ϕ (nonD by R)

OVI(3),OVJ(3), OVK(3)	Same as above but for $\phi - \Delta\phi$
D(3,25,97)	$\vec{D}(\phi,\delta)$ - x,y,z components of distortion vector at point P_ℓ (radial station ℓ_{ND}, azimuth angle ϕ, and age δ) (nonD by R)
QMAT(6,25,25)	$\dfrac{q_z(\eta,\psi,\phi)}{\gamma_{bv}(\phi)}$ - Influence coefficient similar to QTAN but for z component of induced velocity (nonD by tip speed)

E.6.11 COMMON/SVS/

Used in: QVS, SFX

DX,DY,DZ	Δx, Δy, Δz - $\Delta x = x_b - x_a$, $\Delta y = y_b - y_a$, $\Delta z = z_b - z_a$
DXYS	$\Delta x^2 + \Delta y^2$
CL,CLS	$\lvert\vec{c}\rvert$, $\lvert\vec{c}\rvert^2$ - $\vec{c} = \vec{b} - \vec{a}$, $\lvert\vec{c}\rvert^2 = \Delta x^2 + \Delta y^2 + \Delta z^2$
CA,CB	c_a, c_b
ZI	z_x
RAD	$\sqrt{(\Delta x^2 + \Delta y^2) - z_x^2 \lvert\vec{c}\rvert^2}$
ZID	$z_x \Delta z$
XI,YI	x_x, y_x
XK,YK,ZK	x_z, y_z, z_z
CAN	$c_a/\lvert\vec{c}\rvert$
XH,YH,ZH	x_h, y_h, z_h
X1	x_1
Z,ZZ	$z_\perp$, $z_\perp^2$
X2,Z2S	x_2, x_2^2
X1S	x_1^2

CAS,CBS	C_a^2, C_b^2
SOA	$S_{1a} = \sqrt{X_1^2 + Z_\perp^2 + C_a^2}$
SOB	$S_{1b} = \sqrt{X_1^2 + Z_\perp^2 + C_b^2}$
STA	$S_{2a} = \sqrt{X_2^2 + Z_\perp^2 + C_a^2}$
STB	$S_{2b} = \sqrt{X_2^2 + Z_\perp^2 + C_b^2}$
I1	I_1
I2	I_2

APPENDIX F

LISTING OF LDS-73

This program is normally compiled and executed on an IBM 370/168 running under release 21.7 of OS/MVT (Operating System/Multiprogramming with a variable number of tasks) with release 21.6 of FORTRAN G1. The core storage requirement is 180k. The subroutine ERASE, which is used, but not listed, is an MIT library routine which sets all elements of an array to zero.

```
C     1973 AIRLOADS COMPUTATION                                         MAIN0001
C     DIMENSIONED FOR 6 BLADES, 3 EL, 6 ETA, 25 PSI, 97 PHI, AND NH=9   MAIN0002
      REAL LN,LAMP(9,6),LPHS(9,6),UTQ(6,25)                             MAIN0003
      EQUIVALENCE (SUM(1),LAMP(1)),(SUM(100),LPHS(1))                   MAIN0004
      COMMON/SQVLS/I,J,K,KP,KPHI,L,IZ,                                  MAIN0005
     1 XA,YA,ZA,XB,YB,ZB,ZM,VSM,DELL,GA,QT,QK,RHVS,DAM,QMO              MAIN0006
      COMMON/BURST/IVLAC,IVCM,                                          MAIN0007
     1 DVB,FVB,ETAC,DLXY,ZC,RHOC,RHOG,ABOER,ETA,AO,HRS,GK               MAIN0008
      COMMON/SINCO/S(97),C(97),UTS(6,25)                                MAIN0009
      COMMON/SGAM/NE,  NPSI, NPS,NG,   WF,EMU,A,BOER,DANG,              MAIN0010
     1 SUM(6,25,18),ETG(6),DETA(6),ANG(25),PSI(25),ALPHA(6,25)          MAIN0011
      COMMON/SBVGS/IG,MEP,GZER,EL(3),BV(25),GS(25)                      MAIN0012
      COMMON/SQLDS/IABC,IEPS,IS,KZAD,NB,NL,ND,NDEL,NPOP,NPRP,NRNG,NDLM, MAIN0013
     1 FTD,RHOK,DPHR,TANMU,AMBDA,                                       MAIN0014
     2 ZETA(6),KPB(25),ELT(25),ELI(25),FTO(25)                          MAIN0015
      COMMON/SAMB/NH,AMB(25,6),  AMBO(6),CLAM(9,6),SLAM(9,6)            MAIN0016
      COMMON/SLDS/IRB,IRL,IPL,IPRF,IUT,NI,                              MAIN0017
     1 CTO,SM,TH1,VT,EPS,LN,F,CON,ALPS,COSI,GMA,ELO,OMEPS,              MAIN0018
     2 FMT,EMS,DENTH,A1MU,AZERO,AONE,BONE,ATWO,BTWO,THETAO,             MAIN0019
     3 ETE(6),TC(6),ELDS(6),TW(6),GMC(6,25)                             MAIN0020
      COMMON/PLT/IPLX,DEL,ELMN(6),XLDS(6,25)                            MAIN0021
      COMMON/SFX/PZAP,GPS,QTAN(6,25,25),                                MAIN0022
     1 VI(3),VJ(3),VK(3),OVI(3),OVJ(3),OVK(3),D(3,25,97),QMAT(6,25,25)  MAIN0023
      NAMELIST/GENDAT/EMU,AMBDA,EM,RHOK,DPSI,AO,BOER,TANMU,GZER,EPSS,   MAIN0024
     1 CTO,SM,A,TH1,RHOG,WF,VT,DEL,ELMN,QMO,ETG,DETA,EL,DVB,FVB,        MAIN0025
     1 ELI,ELT,FTO,FTD,VSM,RHVS,DAM                                     MAIN0026
     2 /OPTCOU/NB,NE,NL,NC,NI,NG,IG,ND,NH,NQ,IS,IEPS,IVLAC,IABC,IVCM,   MAIN0027
     3 IRL,IRB,IPRF,IOQM,IPGM,IPL,IPLX,IUTQ,MEP,NPOP,NPRP,NRNG,NDLM     MAIN0028
      EXTERNAL FXSQRT                                                   MAIN0029
      CALL ERRSET (207,0,0,0,FXSQRT)                                    MAIN0030
      CALL ERRSET (208,0,0,0,FXSQRT)                                    MAIN0031
      CALL ERRSET (209,0,0,0,FXSQRT)                                    MAIN0032
      CALL ERRSET (251,0,0,0,FXSQRT)                                    MAIN0033
      DVB=0.001                                                         MAIN0034
      FVB=2.001                                                         MAIN0035
C     START OF EACH CASE                                                MAIN0036
```

```
C  READ INPUT DATA                                                      MAIN0037
1     READ (5,GENDAT)                                                   MAIN0038
      READ (5,OPTCOU)                                                   MAIN0039
      CALL TIMING (ITDAT)                                               MAIN0040
C  DEFINE CONSTANTS                                                     MAIN0041
      NLB=NL*NB                                                         MAIN0042
      NPS=360.1/DPSI                                                    MAIN0043
      NDEL=IFIX(EM)*NPS+1                                               MAIN0044
      NPSI=NPS+1                                                        MAIN0045
      EPOP=NPOP                                                         MAIN0046
      DPHI=DPSI*EPOP                                                    MAIN0047
      DANG=DPSI*.1745329E-1                                             MAIN0048
      DPHR=EPOP*DANG                                                    MAIN0049
      ELO=EL(1)                                                         MAIN0050
      GPS=2.0*BOER*EPSS                                                 MAIN0051
      COSI=SQRT(1.0/(1.0+(TANMU/EMU)**2))                               MAIN0052
      ABOER=A*BOER                                                      MAIN0053
      IO=-1                                                             MAIN0054
      IUT=IUTQ                                                          MAIN0055
      IF (ND.EQ.0) GO TO 6                                              MAIN0056
      IF (ND.GT.NL) GO TO 5                                             MAIN0057
C  INPUT  WAKE  DISTORTION                                              MAIN0058
      NDL=NDEL                                                          MAIN0059
      IF (NDEL.GT.NDLM) NDL=NDLM                                        MAIN0060
      DO 195 KP=1,NDL                                                   MAIN0061
195   READ (5,19) ((D(I,J,KP),I=1,3),J=1,NPSI)                          MAIN0062
19    FORMAT (6E13.6)                                                   MAIN0063
5     ND=NL                                                             MAIN0064
      PRINT 1950,IABC                                                   MAIN0065
1950  FORMAT ('1LOADS  PROGRAM  1973    DISTORTED TIP VORTEX MODEL  -   MAIN0066
     1 CASE =',I2)                                                      MAIN0067
      GO TO 1952                                                        MAIN0068
6     CONTINUE                                                          MAIN0069
      PRINT 1951,IABC                                                   MAIN0070
1951  FORMAT ('1LOADS  PROGRAM  1973        RIGID TIP VORTEX MODEL  -   MAIN0071
     1 CASE =',I2)                                                      MAIN0072
```

```
 1952 CONTINUE                                                          MAIN0073
      IF (IPLX.LE.0) GO TO 1002                                         MAIN0074
      IF (IRL.EQ.2) GO TO 1002                                          MAIN0075
C  INPUT  EXPERIMENTAL  AIRLOADS  FOR  PLOT                             MAIN0076
      DO 1000 K=1,NE                                                    MAIN0077
 1000 READ (5,1001)   (XLDS(K,J),J=1,NPSI)                              MAIN0078
 1001 FORMAT (10F7.3)                                                   MAIN0079
 1002 CONTINUE                                                          MAIN0080
C  DEFINE PSI IN DEGREES AND RADIANS PLUS SIN(PSI) & COS(PSI)           MAIN0081
      ANG(1)=0.                                                         MAIN0082
      PSI(1)=0.0                                                        MAIN0083
      C(1)=1.0                                                          MAIN0084
      S(1)=0.0                                                          MAIN0085
      CD= COS(DANG)                                                     MAIN0086
      SD= SIN(DANG)                                                     MAIN0087
      DO 7 J=2,NPSI                                                     MAIN0088
      JG=J-1                                                            MAIN0089
      PSI(J)=PSI(JG)+DANG                                               MAIN0090
      ANG(J)=ANG(JG)+DPSI                                               MAIN0091
      C(J)=CD*C(JG)-SD*S(JG)                                            MAIN0092
 7    S(J)=CD*S(JG)+SD*C(JG)                                            MAIN0093
C  PRINT INPUT DATA                                                     MAIN0094
      CALL PNTIN(ILPT,IPLX,NC,NH,NO,DPSI,EM,EPSS,VSM,RHVS,DAM)          MAIN0095
C  DEFINE ZETA                                                          MAIN0096
      ZETA(1)=0.0                                                       MAIN0097
      ZADD=6.2831853/NB                                                 MAIN0098
      KZAD=ZADD/DANG+0.0001                                             MAIN0099
      DO 8 I=2,NB                                                       MAIN0100
 8    ZETA(I)=ZETA(I-1)+ZADD                                            MAIN0101
C  DEFINE GLAUERT MACH NO. CORRECTION                                   MAIN0102
      DO 9 K=1,NE                                                       MAIN0103
      DO 9 J=1,NPSI                                                     MAIN0104
      UTS(K,J)=ETG(K)+EMU*S(J)                                          MAIN0105
      UTQ(K,J)=0.0                                                      MAIN0106
      GM=SQRT(1.0-(VT*UTS(K,J)/1100.)**2)                               MAIN0107
      IF (GM.LT.0.6) GM=0.6                                             MAIN0108
```

```
      GMC(K,J)=GM                                                       MAIN0109
9     CONTINUE                                                          MAIN0110
      IF (IPGM.EQ.0) GO TO 106                                          MAIN0111
      PRINT 104,       (ETG(K),K=1,NE)                                  MAIN0112
      DO 10 J=1,NPSI                                                    MAIN0113
10    PRINT 105,       ANG(J),(GMC(K,J),K=1,NE)                         MAIN0114
104   FORMAT ('0GLAUERT MACH NO. CORRECTION'/'0ETA',10F10.3/' PSI')     MAIN0115
105   FORMAT (1H ,F4.0,10F10.5)                                         MAIN0116
106   PRINT 107,NG                                                      MAIN0117
107   FORMAT ('1START  ITERATION  ',I1)                                 MAIN0118
C  START OF EACH ITERATION                                              MAIN0119
   12 CONTINUE                                                          MAIN0120
C  COMPUTE  CIRCULATION                                                 MAIN0121
      CALL TIMING (ITGAM)                                               MAIN0122
      CALL GAMT                                                         MAIN0123
C  SET SUM(K,J,LB)=0.0 - NOTE CHANGE WHEN DIMENSIONS CHANGE             MAIN0124
      CALL ERASE (SUM,2700)                                             MAIN0125
C  TEST FOR ANOTHER ITERATION                                           MAIN0126
      IF (NI.GT.NG) GO TO 16                                            MAIN0127
      NC=NC-1                                                           MAIN0128
C  TEST FOR ANOTHER CASE                                                MAIN0129
      IF (NC.GT.0) GO TO 1                                              MAIN0130
      STOP                                                              MAIN0131
16    CONTINUE                                                          MAIN0132
      IF (NG.GT.0) F=1.0                                                MAIN0133
      IF (NQ.EQ.NG) IQ=-1                                               MAIN0134
C  TEST FOR DOWNWASH MATRIX COMPUTATION                                 MAIN0135
      IF (IQ.EQ.1) GO TO 20                                             MAIN0136
C  ERASE OLD DOWNWASH MATRIX                                            MAIN0137
      CALL ERASE(QMAT,3750)                                             MAIN0138
      CALL ERASE(QTAN,3750)                                             MAIN0139
      DO 18 J=1,NPSI                                                    MAIN0140
18    KPB(J)=NDEL                                                       MAIN0141
      PRINT 180,QMO                                                     MAIN0142
180   FORMAT ('1PRINT FOR EACH VORTEX SEGMENT WITH Q.GT.QMO=',F5.3)     MAIN0143
C  COMPUTE NEW DOWNWASH MATRIX                                          MAIN0144
```

```
      CALL QLDS                                                         MAIN0145
      IF (IDQM.EQ.0) GO TO 20                                           MAIN0146
      IE=NE-IDQM+1                                                      MAIN0147
C   PRINT DOWNWASH MATRIX                                               MAIN0148
      DO 185 K=IE,NE                                                    MAIN0149
      EK=K-IE+1                                                         MAIN0150
      FK=EK/2.0-(K-IE+1)/2                                              MAIN0151
      IF (FK.GT.0.1) PRINT 201                                          MAIN0152
201   FORMAT (1H1)                                                      MAIN0153
      PRINT 200,        ETG(K),(ANG(J),J=1,NPS)                         MAIN0154
200   FORMAT ('0QMAT  ETA=',F5.3/'0PHI=   ',24F5.0/' PSI')              MAIN0155
      DO 183 J=1,NPSI                                                   MAIN0156
183   PRINT 202,        ANG(J),(QMAT(K,J,KPHI),KPHI=1,NPS)              MAIN0157
202   FORMAT (1X,F4.0,2X,24F5.1)                                        MAIN0158
185   CONTINUE                                                          MAIN0159
C   END DOWNWASH MATRIX COMPUTATION                                     MAIN0160
20    CONTINUE                                                          MAIN0161
      IF (IUTQ.EQ.0) GO TO 23                                           MAIN0162
C   REMOVE OLD UTQ FROM UTS                                             MAIN0163
      DO 21 K=1,NE                                                      MAIN0164
      DO 21 J=1,NPSI                                                    MAIN0165
21    UTS(K,J)=UTS(K,J)-UTQ(K,J)                                        MAIN0166
23    CONTINUE                                                          MAIN0167
C   USE DOWNWASH MATRIX PLUS CIRCULATION DISTRIBUTION TO COMPUTE DOWNWASH MAIN0168
      DO 25 K=1,NE                                                      MAIN0169
      DO 25 J=1,NPSI                                                    MAIN0170
      AMB(J,K)=TANMU                                                    MAIN0171
      UTQ(K,J)=0.0                                                      MAIN0172
      DO 25 KPHI=1,NPS                                                  MAIN0173
      AMB(J,K)=AMB(J,K)+BV(KPHI)*QMAT(K,J,KPHI)                         MAIN0174
      UTQ(K,J)=UTQ(K,J)+BV(KPHI)*QTAN(K,J,KPHI)                         MAIN0175
25    CONTINUE                                                          MAIN0176
      PRINT 260,(ETG(K),K=1,NE)                                         MAIN0177
260   FORMAT ('1TANGENTIAL INDUCED VELOCITY COMPONENT/ETA + MU*SIN(PSI) MAIN0178
     1'/'0PSI   ETA=  ',F5.3,5(12X,F5.3))                               MAIN0179
      DO 261 J=1,NPSI                                                   MAIN0180
```

```
261     PRINT 262,ANG(J),(UTQ(K,J),UTS(K,J),K=1,NE)                         MAIN0181
262     FORMAT (1H ,F4.0,6(F9.4,'/',F7.4))                                  MAIN0182
        IF (IQ.EQ.1) GO TO 264                                              MAIN0183
        PRINT 263                                                           MAIN0184
263     FORMAT ('OLAMBDA  TPP  FROM  QMAT')                                 MAIN0185
        GO TO 266                                                           MAIN0186
264     PRINT 265                                                           MAIN0187
265     FORMAT ('1LAMBDA  TPP  FROM  QMAT')                                 MAIN0188
266     IF (IUTQ.NE.1) GO TO 900                                            MAIN0189
        DO 27 K=1,NE                                                        MAIN0190
        DO 27 J=1,NPSI                                                      MAIN0191
27      UTS(K,J)=UTS(K,J)+UTQ(K,J)                                          MAIN0192
        GO TO 900                                                           MAIN0193
30      CONTINUE                                                            MAIN0194
        IQ=1                                                                MAIN0195
C  ADD UP DOWNWASH (LAMBDA) CONTRIBUTIONS FROM WAKE ELEMENTS GENERATED      MAIN0196
C  AT VARIOUS RADIAL LOCATIONS ON VARIOUS BLADES                            MAIN0197
        DO 50 K=1,NE                                                        MAIN0198
        DO 50 J=1,NPSI                                                      MAIN0199
        AMB(J,K)=TANMU                                                      MAIN0200
        DO 50 LB=1,NLB                                                      MAIN0201
        AMB(J,K)=AMB(J,K)+SUM(K,J,LB)                                       MAIN0202
50      CONTINUE                                                            MAIN0203
848     CONTINUE                                                            MAIN0204
C  PRINT DOWNWASH DISTRIBUTION                                              MAIN0205
        PRINT 849                                                           MAIN0206
849     FORMAT ('1  DOWNWASH  SUM(K,J,LB)')                                 MAIN0207
        DO 854 K=1,NE                                                       MAIN0208
        EK=K                                                                MAIN0209
        FK=(EK-2.0)/2.0-(K-2)/2                                             MAIN0210
        IF (FK.GT.0.1) PRINT 849                                            MAIN0211
        PRINT 855,       ETG(K),(ZETA(I),I=1,NB)                            MAIN0212
        PRINT 8555,       ((EL(L),L=1,3),I=1,NB)                            MAIN0213
        DO 854 J=1,NPSI                                                     MAIN0214
        PRINT 856,       ANG(J),(SUM(K,J,LB),LB=1,NLB)                      MAIN0215
854     CONTINUE                                                            MAIN0216
```

```
855     FORMAT ('OETA=',F4.3,' ZETA=',F3.1,5(17X,F4.2))                    MAIN0217
8555    FORMAT (' PSI EL=',F4.3,17(F6.3,1X))                               MAIN0218
856     FORMAT (1H ,F4.0,18(1X,F6.4))                                      MAIN0219
        PRINT 600                                                          MAIN0220
600     FORMAT ('1LAMBDA  TPP  FROM  SUM')                                 MAIN0221
900     PRINT 601,(ETG(I),I=1,NE)                                          MAIN0222
601     FORMAT ('OPSI  ETA=',F5.3,9F11.3)                                  MAIN0223
        DO 550 J=1,NPSI                                                    MAIN0224
550     PRINT 602,       ANG(J),(AMB(J,I),I=1,NE)                          MAIN0225
602     FORMAT (1H ,F4.0,10F11.5)                                          MAIN0226
        IF (IQ.NE.1) GO TO 30                                              MAIN0227
C  HARMONIC ANALYSIS OF DOWNWASH                                           MAIN0228
        CALL HART                                                          MAIN0229
C  PRINT HARMONICS OF LAMBDA                                               MAIN0230
        PRINT 603,       (ETG(K),K=1,NE)                                   MAIN0231
603     FORMAT ('OHARMONICS  OF  LAMBDA  TPP'/'OETA',9X,F5.3,5(16X,F5.3)/  MAIN0232
     1 'ONH ',6(5X,'COS',7X,'SIN',3X)/)                                    MAIN0233
        PRINT 604,       (AMBO(K),K=1,NE)                                  MAIN0234
604     FORMAT ('  0 ',6(F11.5,10X))                                       MAIN0235
        DO 70 N=1,NH                                                       MAIN0236
70      PRINT 605,       N,(CLAM(N,K),SLAM(N,K),K=1,NE)                    MAIN0237
605     FORMAT (1X,I2,1X,6(F11.5,F10.5))                                   MAIN0238
C  COMPUTE MAGNITUDE AND PHASE OF HARMONICS OF LAMBDA                      MAIN0239
        DO 72 K=1,NE                                                       MAIN0240
        DO 72 N=1,NH                                                       MAIN0241
        LAMP(N,K)= SQRT(SLAM(N,K)**2+CLAM(N,K)**2)                         MAIN0242
        LPHS(N,K)=57.29578*ATAN2(CLAM(N,K),SLAM(N,K))                      MAIN0243
72      CONTINUE                                                           MAIN0244
C  PRINT MAGNITUDE AND PHASE OF HARMONICS OF LAMBDA                        MAIN0245
        PRINT 606                                                          MAIN0246
606     FORMAT ('ONH ',6(3X,'AMPLITUDE  PHASE  ')/)                        MAIN0247
        PRINT 604,       (AMBO(K),K=1,NE)                                  MAIN0248
        DO 74 N=1,NH                                                       MAIN0249
74      PRINT 607,       N,(LAMP(N,K),LPHS(N,K),K=1,NE)                    MAIN0250
607     FORMAT (1X,I2,1X,6(F11.5,F8.1,2X))                                 MAIN0251
        IUT=IUTQ                                                           MAIN0252
```

```
C  COMPUTE BLADE MOTION AND AIRLOADS                                     MAIN0253
      CALL TIMING (ITLDS)                                                MAIN0254
      CALL LOADS                                                         MAIN0255
      CALL TIMING (ITEND)                                                MAIN0256
      IPL=2                                                              MAIN0257
      IRB=2                                                              MAIN0258
79    IF (NI.GT.NG+1) GO TO 82                                           MAIN0259
      IF (ND.EQ.0) GO TO 80                                              MAIN0260
      PRINT 1950,IABC                                                    MAIN0261
      GO TO 81                                                           MAIN0262
80    PRINT 1951,IABC                                                    MAIN0263
81    CONTINUE                                                           MAIN0264
C  PRINT INPUT DATA                                                      MAIN0265
      CALL PNTIN(ILPT,IPLX,NC,NH,NQ,DPSI,EM,EPSS,VSM,RHVS,DAM)           MAIN0266
C  PRINT VORTEX BURST POINTS                                             MAIN0267
      PRINT 810,       (J,J=1,NPSI),(KPB(J),J=1,NPSI)                    MAIN0268
810   FORMAT ('0J=   ',25I5/' KPB=',25I5)                                MAIN0269
82    CONTINUE                                                           MAIN0270
      IF (IUTQ.NE.2) GO TO 88                                            MAIN0271
      IF (IUT.EQ.9) GO TO 88                                             MAIN0272
      DO 87 K=1,NE                                                       MAIN0273
      DO 87 J=1,NPSI                                                     MAIN0274
87    UTS(K,J)=UTS(K,J)+UTQ(K,J)                                         MAIN0275
      IUT=9                                                              MAIN0276
      CALL LOADS                                                         MAIN0277
      GO TO 79                                                           MAIN0278
88    NG=NG+1                                                            MAIN0279
      IG=0                                                               MAIN0280
C  PRINT EXECUTION TIME                                                  MAIN0281
      TDAT=FLOAT(ITDAT)/100.0                                            MAIN0282
      TEND=FLOAT(ITEND)/100.0                                            MAIN0283
      TQM=FLOAT(ITLDS-ITGAM)/100.0                                       MAIN0284
      TLDS=FLOAT(ITEND-ITLDS)/100.0                                      MAIN0285
      TQLD=FLOAT(ITEND-ITGAM)/100.0                                      MAIN0286
      PRINT 700,        NG,TDAT,TEND,TQM,TLDS,TQLD                       MAIN0287
700   FORMAT ('1START  ITERATION  ',I1,'    EXEC  TIMES  (SEC)  ',       MAIN0288
```

```
1 'TDAT=',F6.2,' TEND=',F6.2,' TQM=',F6.2,' TLDS=',F6.2,' TQLDS=',          MAIN0289
2 F6.2)                                                                      MAIN0290
  GO TO 12                                                                   MAIN0291
  END                                                                        MAIN0292
```

```
      SUBROUTINE PNTIN(IPLT,IPLX,NC,NH,NQ,DPSI,EM,EPSS,VSM,RHVS,DAM)         PNTN0001
C  LDS-73                                                                   PNTN0002
C  PRINT INPUT DATA                                                         PNTN0003
      COMMON/BURST/IVLAC,IVCM,                                              PNTN0004
     1 DVB,FVB,ETAC,DLXY,ZC,RHOC,RHOG,ABOER,ETA,AO,HRS,GK                   PNTN0005
      COMMON/SGAM/NE,  NPSI, NPS,NG,   WF,EMU,A,BOER,DANG,                  PNTN0006
     1 SUM(6,25,18),ETG(6),DETA(6),ANG(25),PSI(25),ALPHA(6,25)              PNTN0007
      COMMON/SBVGS/IG,MEP,GZER,EL(3),BV(25),GS(25)                          PNTN0008
      COMMON/SQLDS/IABC,IEPS,IS,KZAD,NB,NL,ND,NDEL,NPOP,NPRP,NRNG,NDLM,     PNTN0009
     1 FTD,RHOK,DPHR,TANMU,AMBDA,                                           PNTN0010
     2 ZETA(6),KPB(25),ELT(25),ELI(25),FTO(25)                              PNTN0011
      COMMON/SLDS/IRB,IRL,IPL,IPRF,IUT,NI,                                  PNTN0012
     1 CTO,SM,TH1,VT,EPS,LN,F,CON,ALPS,COSI,GMA,ELO,OMEPS,                  PNTN0013
     2 EMT,EMS,DENTH,A1MU,AZERO,AONE,BONE,ATWO,BTWO,THETAC,                 PNTN0014
     3 ETE(6),TC(6),ELDS(6),TW(6),GMC(6,25)                                 PNTN0015
      PRINT 100,        EMU,NB,RHOK,DVB,NRNG,NPRP,NPOP,MEP,NQ,              PNTN0016
     1 TANMU,GZER,RHOG,FVB,ND,NC,IVLAC,IPL,IPLX,                            PNTN0017
     2 AMBDA,A,RHVS,VSM,NE,NL,IEPS,IPRF,                                    PNTN0018
     3 CTO,TH1,EPSS,FTD,NI,NG,IABC,IUT,                                     PNTN0019
     4 SM,AO,EM,IG,IS,NDLM,IVCM,                                            PNTN0020
     5 BOER,DAM,DPSI,WF                                                     PNTN0021
100   FORMAT (                                                              PNTN0022
     1 '0MU=     ',F5.3,4X,'BLADES= ',I1,6X,'RHOK=',F8.6,2X,'DVB=',F5.2,    PNTN0023
     1 2X,'N/NPR=',I2,'/',I1,'  NPOP= ',I1,'  MEP/NQ=',I1,'/',I2/           PNTN0024
     2 '0TANMU= ',F6.4,3X,'GZERO= ',F6.4,2X,'RHOG=',F8.6,2X,'FVB=',F5.2,    PNTN0025
     2 2X,'ND/NC= ',I1,'/',I1,'  IVLAC=',I1,'  IPL/X=',I1,'/',I1/           PNTN0026
     3 '0LAMBDA=',F6.4,3X,'DCL/DA=',F5.3,3X,'RHVS=',F8.6,2X,'VSM=',F5.2,    PNTN0027
     3 2X,'NE/NL= ',I1,'/',I1,'  IEPS= ',I1,'  IPRF=',I2/                   PNTN0028
     4 '0CT=     ',F7.5,2X,'TWIST= ',F5.3,3X,'EPSS=',F4.2,6X,'FTD=',F5.2,   PNTN0029
     4 2X,'NI/NG= ',I1,'/',I1,'  IABC= ',I1,'  IUT=',I1/                    PNTN0030
     5 '0SIGMA= ',F6.4,3X,'A-0=   ',F6.4,2X,'M=    ',F3.1,18X,              PNTN0031
     5    'IG/IS= ',I1,'/',I1,'  NDLM=',I2,'  IVCM=',I1/                    PNTN0032
     6 '0B/2R=  ',F6.4,3X,'DAM=',F7.3,3X,'DPSI=',F4.1,6X,'WF=',F6.2)        PNTN0033
      PRINT 101,        (EL(I),I=1,NL)                                      PNTN0034
      PRINT 102,        (ETG(K),K=1,NE)                                     PNTN0035
      PRINT 103,        (DETA(K),K=1,NE)                                    PNTN0036
```

```
      PRINT 1034,        (ANG(J),J=1,NPSI)                                PNTN0037
      PRINT 1031,        (ELI(J),J=1,NPSI)                                PNTN0038
      PRINT 1032,        (ELT(J),J=1,NPSI)                                PNTN0039
      PRINT 1033,        (FTO(J),J=1,NPSI)                                PNTN0040
 101  FORMAT ('OL=',10F7.3)                                               PNTN0041
 102  FORMAT (' ETA= ',10F7.3)                                            PNTN0042
 103  FORMAT (' DETA=',10F7.3)                                            PNTN0043
 1031 FORMAT (' ELI=',25(1X,F4.2))                                        PNTN0044
 1032 FORMAT (' ELT=',25(1X,F4.2))                                        PNTN0045
 1033 FORMAT (' FTO=',25(1X,F4.2))                                        PNTN0046
 1034 FORMAT ('OPSI= ',25(1X,F4.0))                                       PNTN0047
      RETURN                                                              PNTN0048
      END                                                                 PNTN0049
```

```
      SUBROUTINE GAMT
C  LDS-73
C  COMPUTES CIRCULATION DISTRIBUTION FROM GIVEN ALPHA DISTRIBUTION
      DIMENSION IGP(25),ETG(6),DETA(6),BVS(25),GPLT(128),BVG(3,25),
     1 GBVS(6,25),GBVR(6,25)
      EQUIVALENCE (SUM(1),BVG(1)),(SUM(76),GBVS(1)),(SUM(226),GPLT(1))
      COMMON/SGAM/NE,  NPSI, NPS,NG,   WF,EMU,A,BOER,DANG,
     1 SUM(6,25,18),ETP(6),DETP(6),ANG(25),PSI(25),ALPHA(6,25)
      COMMON/SBVGS/IG,MEP,GZER,EL(3),BV(25),GS(25)
      COMMON/SINCO/S(97),C(97),UTS(6,25)
      DATA BLNK,STAR,PLUS/' ','*','+'/
C  COMPUTE CIRCULATION DISTRIBUTION FROM ALPHA (ANGLE OF ATTACK) DIST.
      AP=A/360.0
C  TEST FOR ZEROTH ITERATION
      IF (IG.GT.1)  GO TO 11
C  TEST FOR READ IN ALPHA DISTRIBUTION
      IF (IG.EQ.1) GO TO 101
      DO 99 I=1,NE
      ETG(I)=ETP(I)
99    DETA(I)=DETP(I)
C  TEST FOR OLD CIRCULATION DISTRIBUTION FROM A PREVIOUS ITERATION
      IF (IA.EQ.1) GO TO 104
      GO TO 1035
101   CONTINUE
      DO 102  I=1,NE
C  READ IN ALPHA DISTRIBUTION
102   READ (5,103)  (ALPHA(I,J),J=1,NPSI)
103   FORMAT (5E14.7)
      READ (5,1030)   (ETG(I),I=1,NE)
      READ (5,1030)   (DETA(I),I=1,NE)
1030  FORMAT (10F6.3)
C  NO OLD CIRCULATION DISTRIBUTION AVAILABLE
1035  FW=1.0
      OWF=1.0-WF
      CALL ERASE (GBVR,150)
      DO 1036 J=1,25
```

```
1036  BVS(J)=GZER                                                       GAMT0037
      IA=1                                                              GAMT0038
104   CONTINUE                                                          GAMT0039
C  ONE OR TWO TRAILING VORTEX LINES                                     GAMT0040
C  START PSI LOOP                                                       GAMT0041
     DO 106 J=1,NPSI                                                    GAMT0042
      GAM=0.0                                                           GAMT0043
C  START ETA LOOP                                                       GAMT0044
      DO 105 I=1,NF                                                     GAMT0045
C  NEW BOUND CIRCULATION DISTRIBUTION                                   GAMT0046
      GA=UTS(I,J)*ALPHA(I,J)                                            GAMT0047
C  AVERAGE OLD AND NEW CIRCULATION DISTRIBUTIONS                        GAMT0048
      GA=FW*AP*GA+OWF*GBVR(I,J)                                         GAMT0049
C  STORE OLD CIRCULATION DISTRIBUTION                                   GAMT0050
      GBVS(I,J)=GBVR(I,J)                                               GAMT0051
C  STORE NEW CIRCULATION DISTRIBUTION                                   GAMT0052
      GBVR(I,J)=GA                                                      GAMT0053
      IF(I.LT.MEP) GO TO 105                                            GAMT0054
C  FIND SPANWISE CIRCULATION PEAK FOR I.GE.MEP                          GAMT0055
      IF (GA.LT.GAM) GO TO 105                                          GAMT0056
      IGP(J)=I                                                          GAMT0057
      GAM=GA                                                            GAMT0058
105   CONTINUE                                                          GAMT0059
C  TIP VORTEX CIRCULATION = BOUND CIRCULATION                           GAMT0060
      BV(J)=BOER*GAM                                                    GAMT0061
106   CONTINUE                                                          GAMT0062
      FW=WF                                                             GAMT0063
C  AZIMUTHAL INTERPOLATION OF CIRCULATION PEAKS                         GAMT0064
      DO 1061 J=1,NPS                                                   GAMT0065
      JM2=J-2                                                           GAMT0066
      JM1=J-1                                                           GAMT0067
      JP1=J+1                                                           GAMT0068
      JP2=J+2                                                           GAMT0069
      IF(JM2.LE.0) JM2=JM2+NPSI-1                                       GAMT0070
      IF(JM1.EQ.0) JM1=NPS                                              GAMT0071
      IF(JP2.EQ.NPSI+1) JP2=2                                           GAMT0072
```

```
      IF (BV(J).GE.BV(JM1)) GO TO 1061                                  GAMT0073
      IF (BV(J).GE.BV(JP1)) GO TO 1061                                  GAMT0074
      IF (IABS(IGP(JP1)-IGP(JM1)).NE.1) GO TO 1061                      GAMT0075
      IM1=IGP(JM1)                                                      GAMT0076
      IP1=IGP(JP1)                                                      GAMT0077
      IF(GBVR(IM1,JM1).LE.GBVR(IM1,JM2)) GO TO 1061                     GAMT0078
      IF(GBVR(IM1,JM1).LE.GBVR(IM1,J))   GO TO 1061                     GAMT0079
      IF(GBVR(IP1,JP1).LE.GBVR(IP1,JP2)) GO TO 1061                     GAMT0080
      IF(GBVR(IP1,JP1).LE.GBVR(IP1,J))   GO TO 1061                     GAMT0081
      BV(J)=(BV(JM1)+BV(JP1))/2.0                                       GAMT0082
1061  CONTINUE                                                          GAMT0083
      BV(NPSI)=BV(1)                                                    GAMT0084
C  SHED WAKE CIRCULATION                                                GAMT0085
      DO 107 J=1,NPS                                                    GAMT0086
107   GS(J)=BV(J+1)-BV(J)                                               GAMT0087
      GS(NPSI)=GS(1)                                                    GAMT0088
C  PRINT CIRCULATION DISTRIBUTION                                       GAMT0089
      PRINT 203,      EL(2),(ETG(I),I=1,NE)                             GAMT0090
203   FORMAT ('0BOUND  CIRCULATION    ELB=',F5.3,20X,'GBVR/GBVS'/       GAMT0091
     1 'OPSI   GBV(3)  GS(3)  ETA=',5X,F6.3,5(10X,F6.3),6X,'GLMX/GTMX') GAMT0092
      DO 109 J=1,NPSI                                                   GAMT0093
      BVG(3,J)=BV(J)/BOER                                               GAMT0094
      BVG(1,J)=GS(J)/BOER                                               GAMT0095
      GLMX=GBVR(NE,J)/BVG(3,J)                                          GAMT0096
      IGMX=IGP(J)                                                       GAMT0097
      GTMX=GLMX                                                         GAMT0098
      IF (IGMX.EQ.NE) GO TO 109                                         GAMT0099
      GTMX=1.0-(1.0-GLMX)*(1.0-ETG(IGMX))/(ETG(NE)-ETG(IGMX))           GAMT0100
109   PRINT 204,      ANG(J),BVG(3,J),BVG(1,J),(GBVR(I,J),GBVS(I,J),    GAMT0101
     1 I=1,NE),GLMX,GTMX                                                GAMT0102
204   FORMAT (1X,F4.0,2F8.4,4X,6(F8.4,'/',F7.4),F6.2,'/',F5.2)          GAMT0103
      IF (NPSI.GT.25) RETURN                                            GAMT0104
C  PRINTER PLOT OF TIP VORTEX CIRCULATION                               GAMT0105
      GZ=GZER/40.0                                                      GAMT0106
      NO=NG-1                                                           GAMT0107
      PRINT 1091,NO,NG                                                  GAMT0108
```

```
1091  FORMAT ('OPLOT OF BVG/GZER VS PSI, +=ITERATION',I3,                GAMT0109
     1 ', *=ITERATION',I3/)                                              GAMT0110
      PRINT 1092                                                         GAMT0111
      PRINT 1093                                                         GAMT0112
1092  FORMAT (' BVG/GZ=-0.25      0.0        0.25       0.50       0.75  GAMT0113
     1   1.0        1.25       1.50       1.75       2.0        2.25     2.5 GAMT0114
     20       2.75')                                                     GAMT0115
1093  FORMAT (' PSI I',25('    I'))                                      GAMT0116
      DO 1094 I=1,128                                                    GAMT0117
1094  GPLT(I)=BLNK                                                       GAMT0118
      DO 1096 J=1,NPSI                                                   GAMT0119
      IX=BVS(J)/GZ+16.5                                                  GAMT0120
      IF (IX.LT.1) IX=1                                                  GAMT0121
      IF (IX.GT.128) IX=128                                              GAMT0122
      GPLT(IX)=PLUS                                                      GAMT0123
      I=BVG(3,J)/GZ+16.5                                                 GAMT0124
      IF (I.LT.1) I=1                                                    GAMT0125
      IF (I.GT.128) I=128                                                GAMT0126
      GPLT(I)=STAR                                                       GAMT0127
      PRINT 1095,      ANG(J),(GPLT(K),K=1,128)                          GAMT0128
1095  FORMAT (1X,F4.0,128A1)                                             GAMT0129
      GPLT(I)=BLNK                                                       GAMT0130
      GPLT(IX)=BLNK                                                      GAMT0131
1096  BVS(J)=BVG(3,J)                                                    GAMT0132
      PRINT 1093                                                         GAMT0133
      PRINT 1092                                                         GAMT0134
      RETURN                                                             GAMT0135
C  ZEROTH GUESS - CONSTANT AZIMUTHAL CIRCULATION DISTRIBUTION            GAMT0136
C  SHED WAKE CIRCULATION SHOULD BE ZERO FOR THIS CASE - INSTEAD IT IS    GAMT0137
C  MADE VERY SMALL BUT FINITE TO ALLOW THE COMPUTATION OF QMAT FOR THE   GAMT0138
C  SHED WAKE AS WELL AS THE TRAILING WAKE                                GAMT0139
11    DO 13  J=1,NPSI                                                    GAMT0140
      BGZ=BUER*GZER                                                      GAMT0141
      GS(J)=BGZ*1.0E-10                                                  GAMT0142
      BVS(J)=GZER                                                        GAMT0143
13    BV(J)=BGZ                                                          GAMT0144
```

```
      IA=0                                                              GAMT0145
      RETURN                                                            GAMT0146
      END                                                               GAMT0147
```

```
      SUBROUTINE QLDS                                                   QLDS0001
C     LDS-73                                                            QLDS0002
C     COMPUTE DOWNWASH MATRIX (QMAT)                                    QLDS0003
      INTEGER OKPHI,OKP                                                 QLDS0004
      COMMON/SQVLS/I,J,K,KP,KPHI,L,IZ,                                  QLDS0005
     1 XA,YA,ZA,XB,YB,ZB,ZM,VSM,DELL,GA,QT,QK,RHVS,DAM,QMC              QLDS0006
      COMMON/BURST/IVLAC,IVCM,                                          QLDS0007
     1 DVB,FVB,ETAC,DLXY,ZC,RHOC,RHOG,ABOER,ETA,AO,HRS,GK               QLDS0008
      COMMON/SINCO/S(97),C(97),UTS(6,25)                                QLDS0009
      COMMON/SGAM/NE,  NPSI, NPS,NG,   WF,EMU,A,BOER,DANG,              QLDS0010
     1 SUM(6,25,18),ETG(6),DETA(6),ANG(25),PSI(25),ALPHA(6,25)          QLDS0011
      COMMON/SBVGS/IG,MEP,GZER,EL(3),BV(25),GS(25)                      QLDS0012
      COMMON/SQLDS/IABC,IEPS,IS,KZAD,NB,NL,ND,NDEL,NPOP,NPRP,NRNG,NDLM, QLDS0013
     1 FTD,RHOK,DPHR,TANMU,AMBDA,                                       QLDS0014
     2 ZETA(6),KPB(25),ELT(25),ELI(25),FTO(25)                          QLDS0015
      COMMON/SFX/PZAP,GPS,QTAN(6,25,25),                                QLDS0016
     1 VI(3),VJ(3),VK(3),OVI(3),OVJ(3),OVK(3),D(3,25,97),QMAT(6,25,25)  QLDS0017
      LST=2                                                             QLDS0018
      IF (IABC.EQ.5) LST=1                                              QLDS0019
      IVLAS=IVLAC                                                       QLDS0020
      IF (IEPS.NE.3) GO TO 18                                           QLDS0021
      RHOI=RHOG                                                         QLDS0022
      RHOS=RHOG                                                         QLDS0023
      GO TO 19                                                          QLDS0024
18    RHOI=EL(2)-EL(1)                                                  QLDS0025
      RHOS=0.40*DPHR                                                    QLDS0026
19    CONTINUE                                                          QLDS0027
C     START  AGE  LOOP                                                  QLDS0028
      DO 500 KP=1,NDEL,NPOP                                             QLDS0029
      OKP=KP+NPOP                                                       QLDS0030
      PZAP=DANG*FLOAT(KP-1)                                             QLDS0031
      PZAO=PZAP+DPHR                                                    QLDS0032
      KPH=NPS-KP                                                        QLDS0033
      IF (OKP.LE.NDLM) GO TO 20                                         QLDS0034
C     COMPUTE DISTORTION EXTRAPOLATION                                  QLDS0035
      DELK=FLOAT(KP-NDLM)*DANG*AMBDA                                    QLDS0036
```

```
      ODELK=DELK+DPHR*AMBDA                                             QLDS0037
20    IF (KPH.GE.1) GO TO 21                                            QLDS0038
C  MAKE SURE KPH GE 1                                                   QLDS0039
      KPH=KPH+NPS                                                       QLDS0040
      GO TO 20                                                          QLDS0041
21    CONTINUE                                                          QLDS0042
C  START  PSI  LOOP                                                     QLDS0043
      DO 490 J=1,NPSI                                                   QLDS0044
C  START  ETA  LOOP                                                     QLDS0045
      DO 480 K=1,NE                                                     QLDS0046
      ETA=ETG(K)                                                        QLDS0047
      ETASP=ETA*S(J)                                                    QLDS0048
      ETACP=ETA*C(J)                                                    QLDS0049
C  START  ZETA  LOOP                                                    QLDS0050
      DO 470 I=1,NB                                                     QLDS0051
      KPHI=KPH+J+KZAD*(I-1)+1                                           QLDS0052
22    IF (KPHI.LT.NPSI) GO TO 23                                        QLDS0053
      KPHI=KPHI-NPS                                                     QLDS0054
      GO TO 22                                                          QLDS0055
23    OKPHI=KPHI-NPOP                                                   QLDS0056
      IF (OKPHI.LT.1) OKPHI=OKPHI+NPS                                   QLDS0057
      NKPHI=KPHI+NPOP                                                   QLDS0058
      IF (NKPHI.GT.NPS) NKPHI=NKPHI-NPS                                 QLDS0059
C  SET TIP VORTEX CIRCULATION FACTOR                                    QLDS0060
      FS=FTO(KPHI)+FTD*PZAP                                             QLDS0061
      IF (FS.GT.1.0) FS=1.0                                             QLDS0062
      RHO=RHOK                                                          QLDS0063
      IF (KP.GE.KPB(KPHI)) RHO=RHOG                                     QLDS0064
C  START  L  LOOP                                                       QLDS0065
      DO 460 L=LST,NL                                                   QLDS0066
C  COMBINATION OF L AND I FOR USE IN SUM(K,J,LB)                        QLDS0067
      LB=L+NL*(I-1)                                                     QLDS0068
      LBS=LB-L+1                                                        QLDS0069
C  VECTORS FROM DOWNWASH COMPUTATION POINT TO ENDS OF VORTEX SEGMENT    QLDS0070
      VI(L)=EL(L)*S(KPHI)-ETASP                                         QLDS0071
      OVI(L)=EL(L)*S(OKPHI)-ETASP                                       QLDS0072
```

```
      VJ(L)=EL(L)*C(KPHI)-ETACP+EMU*PZAP                                QLDS0073
      OVJ(L)=EL(L)*C(OKPHI)-ETACP+EMU*PZAO                              QLDS0074
      IF (L.EQ.ND) GO TO 24                                             QLDS0075
C  RIGID WAKE CASE                                                      QLDS0076
      VK(L)=AMBDA*PZAP-AO*(EL(L)-ETA)                                   QLDS0077
      OVK(L)=AMBDA*PZAO-AO*(EL(L)-ETA)                                  QLDS0078
      GO TO 25                                                          QLDS0079
C  DISTORTED  WAKE  CASE                                                QLDS0080
24    IF (OKP.GT.NDLM) GO TO 245                                        QLDS0081
      VI(L)=VI(L)+D(1,KPHI,KP)                                          QLDS0082
      OVI(L)=OVI(L)+D(1,OKPHI,OKP)                                      QLDS0083
      VJ(L)=VJ(L)+D(2,KPHI,KP)                                          QLDS0084
      OVJ(L)=OVJ(L)+D(2,OKPHI,OKP)                                      QLDS0085
      VK(L)=TANMU*PZAP-AO*(EL(L)-ETA)+D(3,KPHI,KP)                      QLDS0086
      OVK(L)=TANMU*PZAO-AO*(EL(L)-ETA)+D(3,OKPHI,OKP)                   QLDS0087
      GO TO 25                                                          QLDS0088
C  EXTRAPOLATION OF DISTORTION                                          QLDS0089
245   VI(L)=VI(L)+D(1,KPHI,NDLM)                                        QLDS0090
      OVI(L)=OVI(L)+D(1,OKPHI,NDLM)                                     QLDS0091
      VJ(L)=VJ(L)+D(2,KPHI,NDLM)                                        QLDS0092
      OVJ(L)=OVJ(L)+D(2,OKPHI,NDLM)                                     QLDS0093
      VK(L)=TANMU*PZAP-AO*(EL(L)-ETA)+D(3,KPHI,NDLM)+DELK               QLDS0094
      OVK(L)=TANMU*PZAO-AO*(EL(L)-ETA)+D(3,OKPHI,NDLM)+ODELK            QLDS0095
25    IF (L.EQ.1) GO TO 45                                              QLDS0096
      IVLAC=IVLAS                                                       QLDS0097
      FT=FS                                                             QLDS0098
      RHOC=RHO                                                          QLDS0099
      IF (L.EQ.NL) GO TO 251                                            QLDS0100
      IF (IABC.EQ.5) GO TO 37                                           QLDS0101
      IF (IABC.EQ.1) GO TO 45                                           QLDS0102
C  THIS IS AN INBOARD TRAILING VORTEX LINE                              QLDS0103
      IVLAC=3                                                           QLDS0104
      FT=-1.0                                                           QLDS0105
      RHOC=RHOI                                                         QLDS0106
C  VORTEX  LINE  CASES                                                  QLDS0107
251   IF (KP.EQ.NDEL) GO TO 26                                          QLDS0108
```

```
C  CONSTANT  STRENGTH  TRAILING  LINE                                       QLDS0109
      GA=FT*BV(KPHI)                                                        QLDS0110
      XA=VI(L)                                                              QLDS0111
      YA=VJ(L)                                                              QLDS0112
      ZA=VK(L)                                                              QLDS0113
      XB=OVI(L)                                                             QLDS0114
      YB=OVJ(L)                                                             QLDS0115
      ZB=OVK(L)                                                             QLDS0116
      CALL QSVL                                                             QLDS0117
      SUM(K,J,LB)=SUM(K,J,LB)+QK*GA                                         QLDS0118
      QMAT(K,J,KPHI)=QMAT(K,J,KPHI)+QK*FT                                   QLDS0119
      QTAN(K,J,KPHI)=QTAN(K,J,KPHI)+QT*FT                                   QLDS0120
C  TEST FOR VORTEX BURSTING                                                 QLDS0121
      IF (IZ.LE.3) GO TO 26                                                 QLDS0122
      IF (K.GT.1) GO TO 26                                                  QLDS0123
      IF (L.NE.NL) GO TO 26                                                 QLDS0124
      IF (IZ.LE.7) GO TO 253                                                QLDS0125
      IF (KP.GE.KPB(KPHI)) GO TO 26                                         QLDS0126
      KPB(KPHI)=KP                                                          QLDS0127
      KPBT=KP                                                               QLDS0128
      GO TO 254                                                             QLDS0129
253   IF (KP+1.GE.KPB(KPHI)) GO TO 26                                       QLDS0130
      DKP=2.0*(DVB-DLXY)+1.0                                                QLDS0131
      KPBT=KP+DKP                                                           QLDS0132
      IF (KPBT.GE.KPB(KPHI)) GO TO 26                                       QLDS0133
      KPB(KPHI)=KPBT                                                        QLDS0134
254   PRINT 255,       I,J,K,KP,KPHI,IZ,                                    QLDS0135
     1 XA,YA,ZA,XB,YB,ZB,ZC,HRS,GK,ETAC,DLXY                                QLDS0136
255   FORMAT ('0IJK=',I1,I3,I2,' KPKPHI=',I2,I3,' IZ=',I2,                  QLDS0137
     1 ' XYZAB=',3F6.3,F8.3,2F6.3,' ZC=',F8.5,' HRS=',F8.2,                 QLDS0138
     2 ' GK=',F5.3,' ETAC=',F5.2,' DLXY=',F5.2)                             QLDS0139
257   CONTINUE                                                              QLDS0140
      IF (NRNG.LE.KPHI) GO TO 26                                            QLDS0141
C  PROPAGATE VORTEX BURSTING AHEAD OF BLADE UP TO KPHI = NRNG               QLDS0142
C  BURSTING PROPAGATES FASTER THAN BLADE BY ONE EXTRA DPHR STEP             QLDS0143
C  FOR EVERY NPRP DPHR STEPS OF BLADE MOTION                                QLDS0144
```

```
      N=1                                                               QLDS0145
      NP=1                                                              QLDS0146
      KPRP=KPBT                                                         QLDS0147
  258 IF (NP.LE.NPRP) GO TO 259                                         QLDS0148
      NP=0                                                              QLDS0149
      KPRP=KPRP-1                                                       QLDS0150
  259 IF (KPRP.LT.KPB(KPHI+N)) KPB(KPHI+N)=KPRP                         QLDS0151
      N=N+1                                                             QLDS0152
      NP=NP+1                                                           QLDS0153
      IF (KPHI+N.LE.NRNG) GO TO 258                                     QLDS0154
   26 IF (L.LT.NL) GO TO 45                                             QLDS0155
      IF (IABC.GT.3) GO TO 35                                           QLDS0156
  265 IF (KP.LE.2) GO TO 29                                             QLDS0157
C  CONSTANT  STRENGTH  SHED  LINE                                       QLDS0158
   27 LG=L-1                                                            QLDS0159
      IF (IS.EQ.2) GO TO 45                                             QLDS0160
      RHOC=RHOS                                                         QLDS0161
      GA=GS(KPHI)                                                       QLDS0162
      XA=VI(L)                                                          QLDS0163
      YA=VJ(L)                                                          QLDS0164
      ZA=VK(L)                                                          QLDS0165
      XB=VI(LG)                                                         QLDS0166
      YB=VJ(LG)                                                         QLDS0167
      ZB=VK(LG)                                                         QLDS0168
      IVLAC=3                                                           QLDS0169
      CALL QSVL                                                         QLDS0170
      IVLAC=IVLAS                                                       QLDS0171
   28 SUM(K,J,LBS)=SUM(K,J,LBS)+QK*GA                                   QLDS0172
      QMAT(K,J,NKPHI)=QMAT(K,J,NKPHI)+QK                                QLDS0173
      QMAT(K,J,KPHI)=QMAT(K,J,KPHI)-QK                                  QLDS0174
      QTAN(K,J,NKPHI)=QTAN(K,J,NKPHI)+QT                                QLDS0175
      QTAN(K,J,KPHI)=QTAN(K,J,KPHI)-QT                                  QLDS0176
      GO TO 45                                                          QLDS0177
   29 IF (KP.EQ.2) GO TO 32                                             QLDS0178
      IF (I.GT.1) GO TO 45                                              QLDS0179
      IF (IEPS.EQ.2) GO TO 45                                           QLDS0180
```

```
C  NEAR  SHED  SHEET  WITH  EPSILION                                     QLDS0181
30      GA=GS(OKPHI)                                                     QLDS0182
31      LG=L-1                                                           QLDS0183
        ZM=.5*(OVK(L)+OVK(LG))                                           QLDS0184
        DELL=0.5*DPHR*(EL(L)+EL(LG))-GPS                                 QLDS0185
        XA=VI(L)-GPS*C(J)                                                QLDS0186
        XB=VI(LG)-GPS*C(J)                                               QLDS0187
        YA=VJ(L)+GPS*S(J)                                                QLDS0188
        YB=VJ(LG)+GPS*S(J)                                               QLDS0189
        ZA=AMBDA*GPS-AO*(EL(L)-ETA)                                      QLDS0190
        ZB=AMBDA*GPS-AO*(EL(LG)-ETA)                                     QLDS0191
        CALL QVS                                                         QLDS0192
        SUM(K,J,LBS)=SUM(K,J,LBS)+QK*GA                                  QLDS0193
        QMAT(K,J,KPHI)=QMAT(K,J,KPHI)+QK                                 QLDS0194
        QMAT(K,J,OKPHI)=QMAT(K,J,OKPHI)-QK                               QLDS0195
        QTAN(K,J,KPHI)=QTAN(K,J,KPHI)+QT                                 QLDS0196
        QTAN(K,J,OKPHI)=QTAN(K,J,OKPHI)-QT                               QLDS0197
        GO TO 45                                                         QLDS0198
32      IF (IEPS.EQ.2) GO TO 27                                          QLDS0199
        IF (I.GT.1) GO TO 27                                             QLDS0200
        GO TO 45                                                         QLDS0201
C  VORTEX  SHEET  CASES                                                  QLDS0202
C  IABC=4 - VS+VL TIP VORTEX, VL INBOARD TRAILING WAKE, VL SHED WAKE     QLDS0203
C  IABC=5 - SAME AS ABOVE BUT VS INBOARD TRAILING WAKE                   QLDS0204
35      IF (FT.GE.0.9999) GO TO 265                                      QLDS0205
C  TIP VORTEX SHEET                                                      QLDS0206
352     GA=(1.0-FT)*BV(KPHI)                                             QLDS0207
        IF (KP.EQ.1) GO TO 36                                            QLDS0208
353     CONTINUE                                                         QLDS0209
        DELL=EL(3)-ELT(KPHI)                                             QLDS0210
        ZA=VK(L)                                                         QLDS0211
        ZB=OVK(L)                                                        QLDS0212
        ZM=0.5*(ZA+ZB)-AO*DELL                                           QLDS0213
        CALL QVS                                                         QLDS0214
355     SUM(K,J,LB)=SUM(K,J,LB)+QK*GA                                    QLDS0215
        QMAT(K,J,KPHI)=QMAT(K,J,KPHI)+QK*(1.0-FT)                        QLDS0216
```

```
      QTAN(K,J,KPHI)=QTAN(K,J,KPHI)+QT*(1.0-FT)                         QLDS0217
      GO TO 265                                                         QLDS0218
36    IF (I.GT.1) GO TO 353                                             QLDS0219
C  NEAR TIP VORTEX SHEET                                                QLDS0220
      X1=EL(3)-ETA                                                      QLDS0221
      X2=ELT(KPHI)-ETA                                                  QLDS0222
      CBTA=EL(3)*DPHR                                                   QLDS0223
361   DX=X1-X2                                                          QLDS0224
      CBTAS=CBTA*CBTA                                                   QLDS0225
      CX1=SQRT(CBTAS+X1*X1)                                             QLDS0226
      CX2=SQRT(CBTAS+X2*X2)                                             QLDS0227
      IF (RHVS.GT.0.0) GO TO 362                                        QLDS0228
      QK=    ALOG(ABS(X2*(CX1-CBTA)/(X1*(CX2-CBTA))))/DX                QLDS0229
      GO TO 364                                                         QLDS0230
362   IF (CBTAS.GE.RHVS) GO TO 363                                      QLDS0231
      RC=SQRT(RHVS-CBTAS)                                               QLDS0232
      QK=(CBTA/(RC*DX))*(ATAN(CX1/RC)-ATAN(CX2/RC))                     QLDS0233
      GO TO 364                                                         QLDS0234
363   CR=SQRT(CBTAS-RHVS)                                               QLDS0235
      QK=(0.5*CBTA/(CR*DX))*ALOG((CX1-CR)*(CX2+CR)/((CX1+CR)*(CX2-CR))) QLDS0236
364   QT=0.0                                                            QLDS0237
      QL=GA*QK                                                          QLDS0238
      DA=ABS(QL/UTS(K,J))                                               QLDS0239
C  TEST FOR DELTA ALPHA .GT. DELTA ALPHA MAX                            QLDS0240
      IF (DA.LT.DAM) GO TO 367                                          QLDS0241
C  REDUCE INDUCED VELOCITY TO MAKE DA = DAM                             QLDS0242
      QK=QK*DAM/DA                                                      QLDS0243
      QT=QT*DAM/DA                                                      QLDS0244
      QL=GA*QK                                                          QLDS0245
366   QTU=GA*QT                                                         QLDS0246
      PRINT 100, I,J,K,KP,KPHI,L,                                       QLDS0247
     1 XA,YA,ZA,XB,YB,ZB,ZM,DELL,DA,QTU,QL                              QLDS0248
100   FORMAT ('0I=',I1,' J=',I2,' K=',I1,' KP=',I2,' KPHI=',I2,         QLDS0249
     1 ' L=',I1,' XYZAB=',3F6.3,F8.3,2F6.3,' ZM=',F6.3,                 QLDS0250
     2 ' DL=',F5.3,' DA=',F5.3,' QT=',F7.5,' QL=',F7.5)                 QLDS0251
      GO TO 368                                                         QLDS0252
```

```
367   IF (ABS(QL).GT.QMO) GO TO 366                                     QLDS0253
368   IF (L.EQ.NL) GO TO 355                                            QLDS0254
      GO TO 375                                                         QLDS0255
C  INBOARD TRAILING VORTEX SHEET                                        QLDS0256
37    IF (KP.EQ.NDEL) GO TO 45                                          QLDS0257
372   GA=-BV(KPHI)                                                      QLDS0258
      IF (KP.EQ.1) GO TO 38                                             QLDS0259
373   DELL=ELI(KPHI)-EL(1)                                              QLDS0260
      ZM=0.50*(VK(1)+OVK(1))                                            QLDS0261
      XA=ELI(KPHI)*S(KPHI)-ETASP                                        QLDS0262
      XB=ELI(KPHI)*S(OKPHI)-ETASP                                       QLDS0263
      YA=ELI(KPHI)*C(KPHI)-ETACP+EMU*PZAP                               QLDS0264
      YB=ELI(KPHI)*C(OKPHI)-ETACP+EMU*PZAO                              QLDS0265
      ZA=AMBDA*PZAP-AO*(ELI(KPHI)-ETA)                                  QLDS0266
      ZB=AMBDA*PZAO-AO*(ELI(KPHI)-ETA)                                  QLDS0267
      CALL QVS                                                          QLDS0268
375   SUM(K,J,LB)=SUM(K,J,LB)+QK*GA                                     QLDS0269
      QMAT(K,J,KPHI)=QMAT(K,J,KPHI)-QK                                  QLDS0270
      QTAN(K,J,KPHI)=QTAN(K,J,KPHI)-QT                                  QLDS0271
      GO TO 45                                                          QLDS0272
C  NEAR INBOARD TRAILING VORTEX SHEET                                   QLDS0273
38    IF (I.GT.1) GO TO 373                                             QLDS0274
      X1=ELI(KPHI)-ETA                                                  QLDS0275
      X2=EL(1)-ETA                                                      QLDS0276
      CBTA=ELI(KPHI)*DPHR                                               QLDS0277
      GO TO 361                                                         QLDS0278
45    CONTINUE                                                          QLDS0279
C  END  L  LOOP                                                         QLDS0280
460    CONTINUE                                                         QLDS0281
C  END  ZETA  LOOP                                                      QLDS0282
470    CONTINUE                                                         QLDS0283
C  END  ETA  LOOP                                                       QLDS0284
480    CONTINUE                                                         QLDS0285
C  END  PSI  LOOP                                                       QLDS0286
490    CONTINUE                                                         QLDS0287
C  END  AGE  LOOP                                                       QLDS0288
```

```
  500       CONTINUE                                                    QLDS0289
           IVLAC=IVLAS                                                  QLDS0290
           RETURN                                                       QLDS0291
           END                                                          QLDS0292
```

```
      SUBROUTINE QSVL                                                   QSVL0001
C  LDS-73                                                               QSVL0002
C  COMPUTE INDUCED VELOCITY OF FINITE STRAIGHT VORTEX LINE SEGMENT      QSVL0003
C  INCLUDES PROVISION FOR M=0 VERSION OF WRJ LIFTING SURFACE THEORY     QSVL0004
      COMMON/SQVLS/I,J,K,KP,KPHI,L,IZ,                                  QSVL0005
     1 XA,YA,ZA,XB,YB,ZB,ZM,VSM,DELL,GA,QT,QK,RHVS,DAM,QMO              QSVL0006
      COMMON/BURST/IVLAC,IVCM,                                          QSVL0007
     1 DVB,FVB,ETAC,DLXY,ZC,RHOC,RHOG,ABOER,ETA,AO,HRS,GK               QSVL0008
      COMMON/SINCO/S(97),C(97),UTS(6,25)                                QSVL0009
      RHOS=RHOC*RHOC                                                    QSVL0010
      IZ=0                                                              QSVL0011
      GK=1.0                                                            QSVL0012
      QF=1.0                                                            QSVL0013
      ADOTB=XA*XB+YA*YB+ZA*ZB                                           QSVL0014
      AC=XA*XA+YA*YA+ZA*ZA                                              QSVL0015
      BC=XB*XB+YB*YB+ZB*ZB                                              QSVL0016
      CLS=AC+BC-2.*ADOTB                                                QSVL0017
      AXBI=YA*ZB-YB*ZA                                                  QSVL0018
      AXBJ=ZA*XB-ZB*XA                                                  QSVL0019
      AXBK=XA*YB-XB*YA                                                  QSVL0020
      AXBS=AXBI**2+AXBJ**2+AXBK**2                                      QSVL0021
      HS=AXBS/CLS                                                       QSVL0022
      IF (HS.LT.1.0E-20) GO TO 40                                       QSVL0023
C  TEST IF VORTEX BURSTING TO BE CONSIDERED                             QSVL0024
5     IF (IVLAC.EQ.3) GO TO 31                                          QSVL0025
      IKP=I*KP                                                          QSVL0026
C  TEST FOR NEAR WAKE CASE                                              QSVL0027
      IF (IKP.EQ.1) GO TO 31                                            QSVL0028
      DX=XB-XA                                                          QSVL0029
      DY=YB-YA                                                          QSVL0030
      DEN=DY*S(J)-DX*C(J)+1.0E-10                                       QSVL0031
      DETA=AXBK/DEN                                                     QSVL0032
      ETAC=ETA+DETA                                                     QSVL0033
C  TEST FOR VORTEX LINE SEGMENT OR ITS EXTENSION PASSING UNDER BLADE    QSVL0034
      IF (ETAC.GT.1.01.OR.ETAC.LT.0.10) GO TO 31                        QSVL0035
      IF (ABS(DY).LT.0.001) GO TO 25                                    QSVL0036
```

```
      DLXY=(YA-DETA*C(J))/DY                                            QSVL0037
      GO TO 26                                                          QSVL0038
25    DLXY=(XA-DETA*S(J))/DX                                            QSVL0039
C   TEST FOR NEAREST END OF VL SEGMENT BETWEEN ONE SEGMENT LENGTH BEHIND QSVL0040
C   THE BLADE AND ONE HALF SEGMENT LENGTH AHEAD OF THE BLADE            QSVL0041
26    IF (DLXY.LT.-1.501.OR.DLXY.GT.1.001) GO TO 31                     QSVL0042
      IZ=IZ+2                                                           QSVL0043
C   ZC = DISTANCE OF VL SEGMENT (OR ITS EXTENSION) BELOW BLADE          QSVL0044
      ZC=ZA+AD*DETA-DLXY*(ZB-ZA)+1.0E-7                                 QSVL0045
265   ZCT=ABS(ZC/FVB)                                                   QSVL0046
      IF (ZCT.GT.RHOC) GO TO 27                                         QSVL0047
C   BLADE PASSED WITHIN FVB * VORTEX CORE RADIUS OF VL                  QSVL0048
C   THEREFORE VORTEX WILL BURST EVENTUALLY                              QSVL0049
      IZ=IZ+4                                                           QSVL0050
      IF (DLXY.LT.DVB) GO TO 27                                         QSVL0051
C   ALL OF VORTEX LINE HAS PASSED UNDER BLADE - BURST VORTEX IMMEDIATELY QSVL0052
      IZ=IZ+4                                                           QSVL0053
      RHOS=RHOG*RHOG                                                    QSVL0054
      RHOC=RHOG                                                         QSVL0055
C   TEST IF WRJ LS TO BE CONSIDERED                                     QSVL0056
27    IF (IVLAC.EQ.2) GO TO 31                                          QSVL0057
      ZCS=ZC*ZC                                                         QSVL0058
C   TEST FOR VORTEX CORE INFLUNCE                                       QSVL0059
      IF (ZCS.GE.RHOS) GO TO 28                                         QSVL0060
C   MODIFY ONLY ZC TO BE USED IN WRJ-LS                                 QSVL0061
      ZCS=RHOS                                                          QSVL0062
      ZC=RHOC                                                           QSVL0063
C   WRJ LIFTING SURFACE THEORY FOR M=0                                  QSVL0064
28    DXYS=DX*DX+DY*DY                                                  QSVL0065
      DXY=SQRT(DXYS)                                                    QSVL0066
      CD=DEN/DXY                                                        QSVL0067
      SD=(DY*C(J)+DX*S(J))/DXY                                          QSVL0068
      H=ABS(2.865*ZC/ABOER)                                             QSVL0069
      R=-2.865*DETA/ABOER                                               QSVL0070
      CA=-SD                                                            QSVL0071
      SA=CD                                                             QSVL0072
```

```
      ANG=ATAN2(SA,CA)                                                  QSVL0073
      IF (ANG.GT.0.0) GO TO 29                                          QSVL0074
      QF=-1.0                                                           QSVL0075
      ANG=-ANG                                                          QSVL0076
      SA=-SA                                                            QSVL0077
29    IF (ANG.GT.1.5708) GO TO 30                                       QSVL0078
      ANG=3.14159-ANG                                                   QSVL0079
      R=-R                                                              QSVL0080
      QF=-QF                                                            QSVL0081
      CA=-CA                                                            QSVL0082
30    R=R*SA                                                            QSVL0083
      STA=2.*SA*CA                                                      QSVL0084
      SANG=.63662*ANG-.99999                                            QSVL0085
      CBZ=1.926/SQRT(1.0+2.56*SA**2)                                    QSVL0086
      CAO=.434+1.21*SANG**2.75                                          QSVL0087
      CAT=.0135-.0051*SA                                                QSVL0088
      CAPZ= 1.555*SANG-.1899*STA                                        QSVL0089
      CAPO=-.9*SANG+1.15*STA                                            QSVL0090
      CZ=1.683+.563*SANG**2.44                                          QSVL0091
      CO=1.417+.547*SANG**2.13                                          QSVL0092
      CT=.91+.108*SANG**2.25                                            QSVL0093
      CPZ=4.9*(1.0+.15*SA)                                              QSVL0094
      CPO=4.15-1.58*EXP(-7.0*SA)                                        QSVL0095
      RC=R+CA                                                           QSVL0096
      CRZ=-SPAN(0,RC,H+CZ,1.0,0.0)                                      QSVL0097
      CRO=CAO*SPAN(1,RC,H+CO,1.0,0.0)                                   QSVL0098
      CRT=-CAT*SPAN(2,RC,H+CT,1.0,0.0)                                  QSVL0099
      CRPZ=CAPZ*SPANP(0,RC,H+CPZ,CBZ)                                   QSVL0100
      CRPO=CAPO*SPANP(1,RC,H+CPO,CBZ)                                   QSVL0101
      CRC=CRZ+CRO+CRT+CRPZ+CRPO                                         QSVL0102
      GG=-ABOER*DETA*CD/(3.14159*(ZCS+(DETA*CD)**2))                    QSVL0103
C  QF - WRJ LS FACTOR WHICH MULTIPLIES INDUCED DOWNWASH QK              QSVL0104
      QF=QF*CRC/GG                                                      QSVL0105
C  START VORTEX CORE COMPUTATIONS                                       QSVL0106
31    HRS=HS/RHOS                                                       QSVL0107
      IF (IVCM.EQ.2) GO TO 32                                           QSVL0108
```

```
C  FIXED WING VORTEX CORE MODEL                                    QSVL0109
      GK=HS/(RHOS+HS)                                              QSVL0110
      GO TO 33                                                     QSVL0111
C  ROTARY WING VORTEX CORE MODEL                                   QSVL0112
32    IF (HS.GT.RHOS) GO TO 325                                    QSVL0113
      GK=0.2*HS/(RHOS+HS)                                          QSVL0114
      GO TO 33                                                     QSVL0115
325   IF (HRS.GT.10000.) GO TO 33                                  QSVL0116
      GK=1.0-0.9/EXP(0.075*(SQRT(HRS)-1.0))                        QSVL0117
33    AL=SQRT(AC)                                                  QSVL0118
      BL=SQRT(BC)                                                  QSVL0119
      Q=GK*(AL+BL)*(1.0-ADOTB/(AL*BL))/AXBS                        QSVL0120
C  INDUCED DOWNWASH                                                QSVL0121
      QK=Q*AXBK*QF                                                 QSVL0122
C  TANGENTIAL COMPONENT OF INDUCED VELOCITY                        QSVL0123
      QT=Q*(AXBJ*S(J)-AXBI*C(J))                                   QSVL0124
      QL=GA*QK                                                     QSVL0125
      QM=ABS(GA*QT)+ABS(QL)                                        QSVL0126
      DA=ABS(QL/UTS(K,J))                                          QSVL0127
C  TEST FOR DELTA ALPHA .GT. DELTA ALPHA MAX                       QSVL0128
      IF (DA.LT.DAM) GO TO 41                                      QSVL0129
C  REDUCE INDUCED VELOCITY TO MAKE DA = DAM                        QSVL0130
      QK=QK*DAM/DA                                                 QSVL0131
      QT=QT*DAM/DA                                                 QSVL0132
      QL=GA*QK                                                     QSVL0133
35    QTO=GA*QT                                                    QSVL0134
      PRINT 100,    I,IZ,J,K,KP,KPHI,L,RHOC,                       QSVL0135
     1 XA,YA,ZA,XB,YB,ZB,HRS,ZC,DA,                                QSVL0136
     2 GK,QF,CRC,ANG,ETAC,DLXY,GA,Q,QK,QTO,QL                      QSVL0137
100   FORMAT ('0I=',I1,' IZ=',I2,' J=',I2,' K=',I1,' KP=',I2,' KPHI=', QSVL0138
     1 I2,' L=',I1,' RHOC=',F8.6,                                  QSVL0139
     2 ' XYZAB=',3F6.3,F8.3,2F6.3,' HRS=',F8.2,' ZC=',F8.5,' DA=',F5.3/ QSVL0140
     3 ' GK=',F5.3,' QF=',F6.2,' CRC=',F7.3,' ANG=',F6.3,          QSVL0141
     4 ' ETAC=',F6.2,' DLXY=',F6.2,' GA=',F8.5,' Q=',F6.2,         QSVL0142
     5 ' QK=',F6.2,' QT=',F7.5,' QL=',F7.5)                        QSVL0143
      RETURN                                                       QSVL0144
```

```
 40      QK=0.0                                                          QSVL0145
         QT=0.0                                                          QSVL0146
         RETURN                                                          QSVL0147
 C   TEST FOR QMO PRINTED OUTPUT                                         QSVL0148
 41      IF (ABS(QM).GT.QMO) GO TO 35                                    QSVL0149
         RETURN                                                          QSVL0150
        END                                                              QSVL0151
```

```
      FUNCTION SPAN(N,R,H,D1,D2)                                        SPAN0001
C  LDS-73                                                               SPAN0002
      R2=R*R                                                            SPAN0003
      H2=H*H                                                            SPAN0004
      IF (N.GT.1) GO TO 2                                               SPAN0005
      IF (N.EQ.1) GO TO 1                                               SPAN0006
      SPAN=(-R*D1+H*D2)/(R2+H2)                                         SPAN0007
      RETURN                                                            SPAN0008
1     SPAN=2.0*(R*D1*(-R2+3.0*H2)-H*D2*(-3.0*R2+H2))/(R2+H2)**3         SPAN0009
      RETURN                                                            SPAN0010
2     R4=R2*R2                                                          SPAN0011
      RH2=R2*H2                                                         SPAN0012
      H4=H2*H2                                                          SPAN0013
      SPAN=24.0*(R*D1*(-R4+10.0*RH2-5.0*H4)-                            SPAN0014
     1H*D2*(-5.0*R4+10.0*RH2-H4))/(R2+H2)**5                            SPAN0015
      RETURN                                                            SPAN0016
      END                                                               SPAN0017
```

```
      FUNCTION SPANP(N,R,H,B)                                           SPNP0001
C  LDS-73                                                               SPNP0002
      R1=-R*R+H*H+B*B                                                   SPNP0003
      D=R1*R1+4.0*(R*H)**2                                              SPNP0004
      IF (N.EQ.1) GO TO 1                                               SPNP0005
      SPANP=-B*R1/D                                                     SPNP0006
      RETURN                                                            SPNP0007
1     SPANP=2.0*B*(1.0-4.0*R*R*(-R1+4.0*H*H)/D                          SPNP0008
     1 -R1*((-2.0*R*R1+8.0*R*H*H)/D)**2                                 SPNP0009
     2 +R1*(-R1+2.0*R*R+4.0*H*H)/D)/D                                   SPNP0010
      RETURN                                                            SPNP0011
      END                                                               SPNP0012
```

```
      SUBROUTINE QVS                                                    QVS 0001
C     LDS-73                                                            QVS 0002
C     COMPUTE INDUCED VELOCITY OF RECTANGULAR, PLANAR VORTEX SHEET      QVS 0003
      COMMON/SQVLS/I,J,K,KP,KPHI,L,IZ,                                  QVS 0004
     1 XA,YA,ZA,XB,YB,ZB,ZM,VSM,DELL,GA,QT,QK,RHVS,DAM,QMO              QVS 0005
      COMMON/SINCO/S(97),C(97),UTS(6,25)                                QVS 0006
      COMMON/SVS/  DX,DY,DZ,DXYS,CLS,CL,CA,CB,ZI,RAD,ZID,XI,YI,XK,YK,   QVS 0007
     1 ZK,CAN,XH,YH,ZH,X1,Z,ZZ,X2,X2S,X1S,CAS,CBS,SOA,SOB,STA,STB,      QVS 0008
     2 I1,I2                                                            QVS 0009
      REAL I1,I2,IOA,IOB,ITA,ITB                                        QVS 0010
      IZ=0                                                              QVS 0011
      DX=XB-XA                                                          QVS 0012
      DY=YB-YA                                                          QVS 0013
      DZ=ZB-ZA                                                          QVS 0014
      DXYS=DX*DX+DY*DY                                                  QVS 0015
      CLS=DXYS+DZ*DZ                                                    QVS 0016
      CL=SQRT(CLS)                                                      QVS 0017
      CA=-(XA*DX+YA*DY+ZA*DZ)/CL                                        QVS 0018
      CB=CL-CA                                                          QVS 0019
      ZI=(.5*(ZA+ZB)-ZM)/DELL                                           QVS 0020
      RAD=SQRT(DXYS-ZI*ZI*CLS)                                          QVS 0021
      ZID=ZI*DZ                                                         QVS 0022
      XI=(DY*RAD-DX*ZID)/DXYS                                           QVS 0023
      YI=(-DX*RAD-DY*ZID)/DXYS                                          QVS 0024
      XK=(YI*DZ-ZI*DY)/CL                                               QVS 0025
      YK=(ZI*DX-XI*DZ)/CL                                               QVS 0026
      ZK=(XI*DY-YI*DX)/CL                                               QVS 0027
      CAN=CA/CL                                                         QVS 0028
      XH=XA+DX*CAN                                                      QVS 0029
      YH=YA+DY*CAN                                                      QVS 0030
      ZH=ZA+DZ*CAN                                                      QVS 0031
      X1=XI*XH+YI*YH+ZI*ZH                                              QVS 0032
      Z=XK*XH+YK*YH+ZK*ZH                                               QVS 0033
      ZZ=Z*Z                                                            QVS 0034
      X2=X1-DELL                                                        QVS 0035
      X2S=X2*X2                                                         QVS 0036
```

```
      TDELL=(VSM*DELL)**2                                               QVS 0037
C  TESTS FOR TRYING TO COMPUTE Q TOO CLOSE TO EDGE OF VORTEX SHEET      QVS 0038
      IF (X2S.LT.TDELL) GO TO 4                                         QVS 0039
      X1S=X1*X1                                                         QVS 0040
      IF (X1S.LT.TDELL) GO TO 5                                         QVS 0041
 3    CAS=CA*CA                                                         QVS 0042
      CBS=CB*CB                                                         QVS 0043
      RHS=RHVS*RHVS                                                     QVS 0044
      SOA=SQRT(X1S+ZZ+CAS)                                              QVS 0045
      SOB=SQRT(X1S+ZZ+CBS)                                              QVS 0046
      STA=SQRT(X2S+ZZ+CAS)                                              QVS 0047
      STB=SQRT(X2S+ZZ+CBS)                                              QVS 0048
      ZRS=ZZ+RHS                                                        QVS 0049
      ZR=SQRT(ZRS)                                                      QVS 0050
      CRAS=CAS-RHS                                                      QVS 0051
      CRBS=CBS-RHS                                                      QVS 0052
      IF (RHS.GT.CAS) GO TO 6                                           QVS 0053
      CRA=SQRT(CRAS)                                                    QVS 0054
      IOA=(Z*CA/(ZR*CRA))*(ATAN(X2*CRA/(STA*ZR))-ATAN(X1*CRA/(SOA*ZR))) QVS 0055
      ITA=(0.5*CA/CRA)*ALOG((SOA-CRA)*(STA+CRA)/((SOA+CRA)*(STA-CRA)))  QVS 0056
 34   IF (RHS.GT.CBS) GO TO 7                                           QVS 0057
      CRB=SQRT(CRBS)                                                    QVS 0058
      IOB=(Z*CB/(ZR*CRB))*ATAN(ZR*CRB*(X2*SOB-X1*STB)/                  QVS 0059
     1 (SOB*STB*ZRS+X1*X2*CRBS))                                        QVS 0060
      ITB=(0.5*CB/CRB)*ALOG((SOB-CRB)*(STB+CRB)/((SOB+CRB)*(STB-CRB)))  QVS 0061
 36   I1=(IOA+IOB)/DELL                                                 QVS 0062
      I2=(ITA+ITB)/DELL                                                 QVS 0063
      QI=I1*XI+I2*XK                                                    QVS 0064
      QJ=I1*YI+I2*YK                                                    QVS 0065
      QK=I1*ZI+I2*ZK                                                    QVS 0066
      QT=QJ*S(J)-QI*C(J)                                                QVS 0067
      QL=GA*QK                                                          QVS 0068
      QM=ABS(GA*QT)+ABS(QL)                                             QVS 0069
      DA=ABS(QL/UTS(K,J))                                               QVS 0070
C  TEST FOR DELTA ALPHA .GT. DELTA ALPHA MAX                            QVS 0071
      IF (DA.LT.DAM) GO TO 8                                            QVS 0072
```

```
C  REDUCE INDUCED VELOCITY TO MAKE DA = DAM                                QVS 0073
      QK=QK*DAM/DA                                                         QVS 0074
      QT=QT*DAM/DA                                                         QVS 0075
      QL=GA*QK                                                             QVS 0076
35    QTO=GA*QT                                                            QVS 0077
      PRINT 100, I,J,K,KP,KPHI,L,                                          QVS 0078
     1 XA,YA,ZA,XB,YB,ZB,ZM,DELL,DA,QTO,QL,                                QVS 0079
     2 RHVS,CA,CB,X1,X2,Z,XI,YI,ZI,XK,YK,ZK,IOA,IOB,ITA,ITB                QVS 0080
100   FORMAT ('0I=',I1,' J=',I2,' K=',I1,' KP=',I2,' KPHI=',I2,            QVS 0081
     1 ' L=',I1,' XYZAB=',3F6.3,F8.3,2F6.3,' ZM=',F6.3,                    QVS 0082
     2 ' DL=',F5.3,' DA=',F5.3,' QT=',F7.5,' QL=',F7.5/                    QVS 0083
     3 ' RHVS=',F5.4,' CAB=',2F6.3,' X12=',2F6.3,' Z=',F6.3,               QVS 0084
     4 ' XYZIK=',3F6.3,F8.3,2F6.3,' IOABT=',2F6.3,F8.3,F6.3)               QVS 0085
      RETURN                                                               QVS 0086
C  EXTEND EDGE OF VORTEX SHEET                                             QVS 0087
C  TO AVOID COMPUTING Q WITHIN VSM OF FRONT OR BACK EDGES                  QVS 0088
4     X2S=TDELL                                                            QVS 0089
      IZ=2                                                                 QVS 0090
      X0=X2                                                                QVS 0091
      X2=SIGN(VSM*DELL,X2)                                                 QVS 0092
      X1=X2+DELL                                                           QVS 0093
      X1S=X1*X1                                                            QVS 0094
      GO TO 3                                                              QVS 0095
5     X1S=TDELL                                                            QVS 0096
      IZ=1                                                                 QVS 0097
      X0=X1                                                                QVS 0098
      X1=SIGN(VSM*DELL,X1)                                                 QVS 0099
      X2=X1-DELL                                                           QVS 0100
      X2S=X2*X2                                                            QVS 0101
      GO TO 3                                                              QVS 0102
6     RCAS=-CRAS                                                           QVS 0103
      RCA=SQRT(RCAS)                                                       QVS 0104
      IOA=(0.5*CA*Z/(ZR*RCA))*ALOG((ZR*SOA-X1*RCA)*(ZR*STA+X2*RCA)/         QVS 0105
     1 ((ZR*SOA+X1*RCA)*(ZR*STA-X2*RCA)))                                  QVS 0106
      ITA=(CA/RCA)*(ATAN(SOA/RCA)-ATAN(STA/RCA))                           QVS 0107
      GO TO 34                                                             QVS 0108
```

```
 7      RCBS=-CRBS
        RCB=SQRT(RCBS)
        IOB=(0.5*CB*Z/(ZR*RCB))*ALOG((ZR*SOB-X1*RCB)*(ZR*STB+X2*RCB)/
      1 ((ZR*SOB+X1*RCB)*(ZR*STB-X2*RCB)))
        ITB=(CB/RCB)*(ATAN(SOB/RCB)-ATAN(STB/RCB))
        GO TO 36
 8       IF (ABS(QM).GT.QMO) GO TO 35
        RETURN
        END
```

```
      SUBROUTINE LOADS                                                  LOAD0001
C  LDS-73                                                               LOAD0002
C  COMPUTE ROTOR AIRLOADS (LB/IN) AS A FUNCTION OF RADIUS AND AZIMUTH   LOAD0003
      REAL L,LN                                                         LOAD0004
      DIMENSION TF(6,25),AF(6,25),HF(6,25),AIRL(6,25),ALPHS(6,25),      LOAD0005
     1 RF(6,25),CTPR(6,25),CDP(6,25),CDMP(6,25),DQP(6,25),DQI(6,25),    LOAD0006
     2 ALPRP(6,25),ALPSS(6,25),EMAP(6,25),EMDP(6,25),DZDR(6,25),        LOAD0007
     3 ELFP(6,25)                                                       LOAD0008
      EQUIVALENCE (SUM(1),TF(1)),(SUM(151),AF(1)),(SUM(301),HF(1)),     LOAD0009
     1 (SUM(451),RF(1)),(SUM(601),CTPR(1)),(SUM(751),CDP(1)),           LOAD0010
     2 (SUM(901),CDMP(1)),(SUM(1051),AIRL(1)),(SUM(1201),ALPHS(1)),     LOAD0011
     3 (SUM(1351),DQP(1)),(SUM(1501),DQI(1)),(SUM(1651),DZDR(1)),       LOAD0012
     4 (SUM(1801),ALPRP(1)),(SUM(1951),ALPSS(1)),(SUM(2101),EMAP(1)),   LOAD0013
     5 (SUM(2251),EMDP(1)),(SUM(2401),ELFP(1))                          LOAD0014
      COMMON/SINCO/S(97),C(97),UTS(6,25)                                LOAD0015
      COMMON/SGAM/NETA,NPSI, NPS,NG,    WF,EMU,A,BDER,DANG,             LOAD0016
     1 SUM(6,25,18),ETG(6),DETA(6),ANG(25),PSI(25),ALPH(6,25)           LOAD0017
      COMMON/SAMB/NH,AMBDA(25,6),AMBO(6),CLAM(9,6),SLAM(9,6)            LOAD0018
      COMMON/SLDS/IRB,IRL,IPL,IPRF,IUT,NI,                              LOAD0019
     1 CT0,SM,TH1,VT,EPS,LN,F,CON,ALPS,COSI,GMA,EL0,OMEPS,              LOAD0020
     2 EMT,EMS,DENTH,AIMU,AZERO,AONE,BONE,ATWO,BTWO,THETAO,             LOAD0021
     3 ETA(6),TC(6),ELDS(6),TW(6),GMC(6,25)                             LOAD0022
      NAMELIST/ROTOR/EPS,LN,ALPS,CON,F,EMT,TC,TW,IL,IPRG,IPH,IPUN       LOAD0023
     1/PERF/IPRF,TH1,TC,TW                                              LOAD0024
      IF (IUT.EQ.9) GO TO 3                                             LOAD0025
      IF (IRL.EQ.2) GO TO 2                                             LOAD0026
C  READ AIRLOADS PARAMETERS                                             LOAD0027
      READ (5,ROTOR)                                                    LOAD0028
      SALP=ALPS                                                         LOAD0029
C  WRITE AIRLOADS PARAMETERS                                            LOAD0030
2     PRINT 100,EPS,LN,TH1,CT0,SM,VT,SALP,CON,F,(TC(I),I=1,NETA)        LOAD0031
100   FORMAT ('1LOADS  INPUTS'/1H0,                                     LOAD0032
     1 '  EPS=',F6.4,'  LN=',F6.3,'  TH1=',F5.3/'0CT=',F6.4,'  SIGMA=', LOAD0033
     2 F6.4,   '  VT=',F5.1,'  ALPS=',F3.0,'  CON=',F8.3,'  F=',F4.2/   LOAD0034
     3 '0T/C=',10F7.3)                                                  LOAD0035
C  DEFINE CONSTANTS                                                     LOAD0036
```

```
3     ALPS=.0174533*SALP
      SMLR=0.5*SM
      CONST=1.0/FLOAT(NPS)
      IGO=IL
      IF (IUT.EQ.9) GO TO 84
      CALL HSL(IGO)
      IF (IL.EQ.3) GO TO 84
C  COMPUTE ANGLE OF ATTACK (ALPHA) AND AIRLOADS DISTRIBUTIONS
C  USE HARMONIC SOLUTION FLAPPING
      DO 40 I=1,NETA
      ELDS(I)=0.0
      DO 30 J=1,NPSI
      CTP=2.*C(J)*C(J)-1.
      STP=2.*S(J)*C(J)
      UT=UTS(I,J)
      UP=AMBDA(J,I)+A1MU+EMU*C(J)*(AZERO-AONE*C(J)-BONE*S(J)-ATWO*CTP-
     1 BTWO*STP)+ETA(I)*(AONE*S(J)-BONE*C(J)+2.*ATWO*STP-2.*BTWO*CTP)
      ALPH(I,J)=THETAO+TW(I)-ATAN(UP/UT)
C  REVERSE FLOW
      IF (UT.LT.0.0) ALPH(I,J)=-ALPH(I,J)
      ALP=ALPH(I,J)
      IF (ALP.GT.0.7853982) ALP=1.5707963-ALP
      IF (ALP.LT.-0.7853982) ALP=-1.5707963-ALP
C  STALL
      IF (ABS(ALP).GT.ALPS) ALP=SIGN(ALPS,ALP)
      AIRL(I,J)=UT*UT*ALP
      ALPHS(I,J)=ALP
30    ELDS(I)=ELDS(I)+AIRL(I,J)
40    ELDS(I)=ELDS(I)-AIRL(I,J)
C  ADD (1.0-F) TIMES ZEROTH HARMONIC AIRLOADS
45    ONFP=(1.0-F)/NPS
      DO 46 I=1,NETA
      DLDS=OFNP*ELDS(I)
      DO 46 J=1,NPSI
      AIRL(I,J)=F*AIRL(I,J)+DLDS
46    CONTINUE
```

```
 50     CONTINUE
C  START DRAG, TORQUE, AND H FORCE COMPUTATIONS
        Q=0.
        T=0.
        H=0.
        QP=0.0
        QI=0.0
C  START ETA LOOP
        DO 80 I=1,NETA
        ETB=ETG(I)
        DELTH=TW(I)
        DELEP=DETA(I)*CONST
        ALO=ALPH(I,NPS)
C  START PSI LOOP
        DO 70 J=1,NPSI
        ALP=ALPH(I,J)
C  DELTA ALPHA/DELTA PSI
        DALP=(ALP-ALO)/DANG
        ALO=ALP
C  CHORDWISE FLOW VELOCITY
        VCH=UTS(I,J)
C  SPANWISE (RADIAL) FLOW VELOCITY
        VSP=EMU*C(J)
C  RESULTANT FLOW VELOCITY
        VR=SQRT(VCH*VCH+VSP*VSP)
C  SWEEP CORRECTION TO CDO
        CDO=0.0087*VR/ABS(VCH)
C  BLADE ELEMENT MACH NUMBER
        EMA=ABS(VCH*EMT)
        EMB=EMA
        IF (EMB.GT.0.60) EMB=0.60
        IF (EMB.LT.0.20) EMB=0.20
C  ASSUME HUB PLANE = TPP
C  COMPUTE PITCH ANGLE OF BLADE ELEMENT RELATIVE TO HUB PLANE
        THE=THETAO+DELTH+BONE*C(J)-AONE*S(J)
C  CORRECT LIFT CURVE SLOPE FOR MACH NUMBER
```

```
      DCLDA=A/GMC(I,J)                                                  LOAD0109
      ALPP=ABS(ALP)                                                     LOAD0110
      IF (ALPP.LT.1.5708) GO TO 54                                      LOAD0111
      ALPP=3.14159-ALPP                                                 LOAD0112
      VCH=-VCH                                                          LOAD0113
      ALP=-SIGN(ALPP,ALP)                                               LOAD0114
C  DYNAMIC STALL PARAMETER                                              LOAD0115
54    DSP=SQRT(ABS(2.0*BOER*DALP/VCH))                                  LOAD0116
C  DELTA STALL ALPHA DUE TO PITCHING VELOCITY                           LOAD0117
      GDS=1.076*ALOG(0.60/EMB)*DSP                                      LOAD0118
      IF (GDS.GT.ALPP) GDS=ALPP                                         LOAD0119
C  REFERENCE ALPHA                                                      LOAD0120
      ALPR=ABS(ALP-SIGN(GDS,DALP))                                      LOAD0121
C  REQUIRE ALPR.GE.ALPP/2                                               LOAD0122
      ALMN=0.50*ALPP                                                    LOAD0123
      IF (ALPR.LT.ALMN) ALPR=ALMN                                       LOAD0124
      IF (ALPR.GT.1.5708) ALPR=1.5708                                   LOAD0125
C  START LIFT COMPUTATION                                               LOAD0126
C  STATIC STALL ALPHA                                                   LOAD0127
      ALSS=0.26-0.0065/(0.65-EMB)                                       LOAD0128
      IF (VCH.LT.0.0) ALSS=0.130                                        LOAD0129
      IF (ALP.LT.0.0) ALSS=0.130                                        LOAD0130
C  STATIC STALL CL                                                      LOAD0131
      CLSS=ALSS*DCLDA                                                   LOAD0132
C  START DETERMINATION OF REFERENCE LIFT CURVE SLOPE                    LOAD0133
      AR=DCLDA                                                          LOAD0134
      IF (ALPR.LE.ALSS) GO TO 58                                        LOAD0135
C  FIND STATIC CL AT ALPR                                               LOAD0136
      IF (ALPR.GE.0.280) GO TO 55                                       LOAD0137
      CLS=CLSS+(1.0-CLSS)*(ALPR-ALSS)/(0.280-ALSS)                      LOAD0138
      GO TO 57                                                          LOAD0139
55    IF (ALPR.GT.0.7854) GO TO 56                                      LOAD0140
      CLS=1.0                                                           LOAD0141
      GO TO 57                                                          LOAD0142
56    CLS=SIN(ALPR)                                                     LOAD0143
C  REFERENCE LIFT CURVE SLOPE - INCLUDING SWEEP EFFECT                  LOAD0144
```

```
57      AR=ABS(CLS*VR/(ALPR*VCH))                                         LOAD0145
        IF (AR.GT.DCLDA) AR=DCLDA                                         LOAD0146
C  LIFT INCLUDING DYNAMIC EFFECTS                                         LOAD0147
58      CLN=ALP*AR                                                        LOAD0148
        ELF=VCH*VCH*CLN                                                   LOAD0149
C  START DRAG COMPUTATION                                                 LOAD0150
C  ACTUAL DRAG IS STATIC DRAG AT ALPHA REFERENCE (ALPR)                   LOAD0151
        IF (ALP.LT.0.0) ALSS=0.070                                        LOAD0152
        IF (ALPR.GT.ALSS) GO TO 60                                        LOAD0153
C  UNSTALLED DRAG                                                         LOAD0154
        CD=CDO-ALPR*(0.0216-0.40*ALPR)                                    LOAD0155
        GO TO 62                                                          LOAD0156
C  STALLED DRAG                                                           LOAD0157
60      CDSS=CDO-ALSS*(0.0216-0.40*ALSS)                                  LOAD0158
        IF (ALPR.GE.0.260) GO TO 61                                       LOAD0159
        CD=CDSS+(0.20-CDSS)*(ALPR-ALSS)/(0.260-ALSS)                      LOAD0160
        GO TO 62                                                          LOAD0161
61      CD=0.20+1.8*(ALPR-0.260)                                          LOAD0162
        IF (CD.GT.2.0) CD=2.0                                             LOAD0163
C  DRAG DIVERGENCE MACH NUMBER                                            LOAD0164
62      EMD=0.85-0.33*ABS(CLN)                                            LOAD0165
        IF (CD.GT.0.018) EMD=EMD-SQRT(CD-0.018)                           LOAD0166
        IF (EMD.LT.0.10) EMD=0.10                                         LOAD0167
        DEM=EMA-EMD                                                       LOAD0168
        CDM=0.0                                                           LOAD0169
        IF (DEM.LE.0.0) GO TO 63                                          LOAD0170
        CDM=DEM*DEM                                                       LOAD0171
        IF (DEM.GT.0.15) CDM=0.0225+0.40*(DEM-0.15)                       LOAD0172
        CD=CD+CDM                                                         LOAD0173
C  AZIMUTHAL DRAG                                                         LOAD0174
63      DA=VCH*VCH*CD                                                     LOAD0175
C  RADIAL DRAG                                                            LOAD0176
        DR=0.0087*VSP*VR                                                  LOAD0177
C  SIN AND COS OF ANGLE BETWEEN LOCAL WIND AND TPP                        LOAD0178
        STA=SIN(ALP-THE)                                                  LOAD0179
        CTA=COS(ALP-THE)                                                  LOAD0180
```

```
C  CORRECTION FOR ANGLE BLADE ELEMENT MAKES WITH TPP IN RADIAL DIRECTION
      SZRL=ELF*SIN(DZDR(I,J))
C  THRUST RELATIVE TO TPP
      TFOR=ELF*CTA+DA*STA
C  AZIMUTHAL FORCE IN TPP
      AFOR=DA*CTA-ELF*STA
C  RADIAL FORCE IN TPP
      RFOR=DR-SZRL
C  HORIZONTAL FORCE IN TPP
      HFOR=RFOR*C(J)+AFOR*S(J)
C  PARISITE TORQUE INCREMENT
      DQP(I,J)=DA*CTA*ETB*DELEP*SMLR*1.0E+6
C  INDUCED TORQUE INCREMENT
      DQI(I,J)=-ELF*STA*ETB*DELEP*SMLR*1.0E+6
C  SUM UP ELEMENTAL FORCES AND TORQUE
      T=T+TFOR*DELEP
      H=H+HFOR*DELEP
      Q=Q+ETB*AFOR*DELEP
      QP=QP+DQP(I,J)
      QI=QI+DQI(I,J)
C  STORE FOR LATER PRINT OUT
      CDP(I,J)=CD
      CDMP(I,J)=CDM
      TF(I,J)=TFOR*CON
      AF(I,J)=AFOR*CON
      HF(I,J)=HFOR*CON
      RF(I,J)=RFOR*CON
      ELFP(I,J)=ELF*CON
      ALPHS(I,J)=57.29578*ALPH(I,J)
      ALPRP(I,J)=57.29578*ALPR
      ALPH(I,J)=57.29578*CLN/A
      ALPSS(I,J)=57.29578*ALSS
      CTPR(I,J)=TFOR*DELEP*SMLR*1.0E+6
      EMAP(I,J)=EMA
      EMDP(I,J)=EMD
      AIRL(I,J)=DCLDA*CON*AIRL(I,J)
```

```
C   END PSI LOOP                                                        LOAD0217
 70     CONTINUE                                                        LOAD0218
C   SUBTRACT OUT LAST INCREMENT TO ALL SUMS - SINCE J=1 SAME AS J=NPSI  LOAD0219
        T=T-TFOR*DELEP                                                  LOAD0220
        H=H-HFOR*DELEP                                                  LOAD0221
        Q=Q-ETB*AFOR*DELEP                                              LOAD0222
        QP=QP-DQP(I,J)                                                  LOAD0223
        QI=QI-DQI(I,J)                                                  LOAD0224
C   END ETA LOOP                                                        LOAD0225
 80     CONTINUE                                                        LOAD0226
C   VEHICLE LIFT/DRAG                                                   LOAD0227
        ELD=EMU*T/Q                                                     LOAD0228
C   NET PROPULSIVE FORCE                                                LOAD0229
        X=T*SQRT(1.0-COSI*COSI)-H*COSI                                  LOAD0230
C   ROTOR LIFT/DRAG                                                     LOAD0231
        ELDR=EMU*T/(Q-EMU*X)                                            LOAD0232
C   COMPUTE THRUST, HORIZONTAL FORCE, AND TORQUE COEF. (CT,CH,CQ)       LOAD0233
        CTF=T*SMLR                                                      LOAD0234
        CH=H*SMLR                                                       LOAD0235
        CQ=Q*SMLR                                                       LOAD0236
        CQP=QP*1.0E-6                                                   LOAD0237
        CQI=QI*1.0E-6                                                   LOAD0238
C   PROPULSIVE FORCE COEF.                                              LOAD0239
        CX=X*SMLR                                                       LOAD0240
C   LIFT & PROPULSIVE FORCE/(Q*D**2*SIGMA)                              LOAD0241
        CLSTR=0.7854*T/EMS                                              LOAD0242
        CXSTR=0.7854*X/EMS                                              LOAD0243
C   PRINT AIRLOADS OUTPUT                                               LOAD0244
        CALL PNTLDS (IPRG)                                              LOAD0245
        PRINT 140,       CTF,CH,CX,CQ,CQP,CQI,ELD,ELDR,CLSTR,CXSTR      LOAD0246
140     FORMAT ('OCT=',F8.6,'  CH=',F8.6,'  CX=',F8.6,'  CQ=',F8.6,     LOAD0247
     1 '  CQP=',F8.6,'  CQI=',F8.6,'  L/D=',F7.3,'  L/DR=',F7.3/        LOAD0248
     2 'OCLSTAR=',F8.6,'  CXSTAR=',F8.6////)                            LOAD0249
        IF (IGO.NE.1) GO TO 90                                          LOAD0250
 84     CONTINUE                                                        LOAD0251
        IGO=2                                                           LOAD0252
```

```
C  FLAPPING SOLUTION INCLUDING ALL FLAPPING HARMONICS                    LOAD0253
C  PLUS ALL HARMONICS OF FIRST BLADE BENDING MODE                        LOAD0254
      CALL BLADE                                                         LOAD0255
      GO TO 50                                                           LOAD0256
90    IF (IUT.EQ.2) RETURN                                               LOAD0257
      IF (IPUN.GT.0.AND.IPUN.GT.NG) GO TO 92                             LOAD0258
C  PUNCH ALPHA DISTRIBUTION FOR GAMMA COMPUTATIONS IN WG71                LOAD0259
      DO 91 I=1,NETA                                                     LOAD0260
      PUNCH 904,ETG(I)                                                   LOAD0261
904   FORMAT (F6.3)                                                      LOAD0262
91    PUNCH 902,    (ALPH(I,J),J=1,NPSI)                                 LOAD0263
902   FORMAT (5E14.7)                                                    LOAD0264
92    CONTINUE                                                           LOAD0265
C  HARMONIC ANALYSIS OF AIRLOADS                                         LOAD0266
      IF (IPH.EQ.1.OR.IPH.LE.NG) CALL HALDS                              LOAD0267
C  CARTESIAN AIRLOADS PLOTS                                              LOAD0268
      IF (IPL.EQ.1.OR.IPL.LE.NG) CALL PLTL                               LOAD0269
      IF (NG.LE.IPRF) RETURN                                             LOAD0270
      READ (5,PERF)                                                      LOAD0271
      RETURN                                                             LOAD0272
      END                                                                LOAD0273
```

```
      SUBROUTINE HSL(IGO)                                               HSL 0001
C     LDS-73                                                            HSL 0002
C     HARMONIC SOLUTION OF FLAPPING EQUATION IN NFP                     HSL 0003
      REAL L,LN                                                         HSL 0004
      COMMON/SGAM/NETA,NPSI, NPS,NG,    WF,EMU,A,BOER,DANG,             HSL 0005
     1 SUM(6,25,18),ETG(6),DETA(6),ANG(25),PSI(25),ALPH(6,25)           HSL 0006
      COMMON/SAMB/NH,AMBDA(25,6),AMBO(6),CLAM(9,6),SLAM(9,6)            HSL 0007
      COMMON/SLDS/IRB,IRL,IPL,IPRF,IUT,NI,                              HSL 0008
     1 CTO,SM,TH1,VT,EPS,LN,F,CON,ALPS,COSI,GMA,ELO,OMEPS,              HSL 0009
     2 EMT,EMS,DENTH,AIMU,AZERO,AONE,BONE,ATWO,BTWO,THETAO,             HSL 0010
     3 ETA(6),TC(6),ELDS(6),TW(6),GMC(6,25)                             HSL 0011
      EMS=EMU*EMU                                                       HSL 0012
      HEMU=.5*EMU                                                       HSL 0013
      HEMS=.5*EMS                                                       HSL 0014
      GMA=SQRT(1.0-(.75*EMT)**2)                                        HSL 0015
      IF (GMA.LT.0.6) GMA=0.6                                           HSL 0016
      CTSA=2.0*CTO/(SM*A)                                               HSL 0017
      AOTP=A/(6.28318*GMA)                                              HSL 0018
C     START HARMONIC FLAPPING SOLUTION                                  HSL 0019
C     DEFINE CONSTANTS                                                  HSL 0020
      OE1=1.-ELO                                                        HSL 0021
      OE2=1.-ELO**2                                                     HSL 0022
      OE3=1.-ELO**3                                                     HSL 0023
      OE4=1.-ELO**4                                                     HSL 0024
      OE5=1.-ELO**5                                                     HSL 0025
      EINZ=.5*(OE2-2.*EPS*OE1)                                          HSL 0026
      EIN1=.333333*(OE3-1.5*EPS*OE2)                                    HSL 0027
      EIN2=.25*(OE4-1.33333*EPS*OE3)                                    HSL 0028
      EIN3=.2*(OE5-1.25*EPS*OE4)                                        HSL 0029
      DENF=EMS*EINZ                                                     HSL 0030
      DIN=EIN2-EPS*EIN1                                                 HSL 0031
      DENT=3.*DENF+4.*DIN                                               HSL 0032
      DEN=OE3+1.5*EMS                                                   HSL 0033
      ENUM=OE2*(1.-.5*EPS)-EPS*OE1                                      HSL 0034
      DENB=DENF+4.*DIN                                                  HSL 0035
      DENTH=DEN-12.*EMS*EIN1*ENUM/DENT                                  HSL 0036
```

```
      FEPSI=((1.-.5*EPS)*EIN1-.5*EPS*EINZ)                              HSL 0037
      HALN=.5*AOTP*LN                                                   HSL 0038
      FLN=F*AOTP*LN/6.                                                  HSL 0039
      EFLN=2.*FLN*(EIN2-EPS*EIN1)                                       HSL 0040
      FTWO=2.*EPS*EINZ-(1.+EPS)*EIN1                                    HSL 0041
      ALZ=0.                                                            HSL 0042
      CLZ=0.                                                            HSL 0043
      ZLZ=0.                                                            HSL 0044
      ALC=0.                                                            HSL 0045
      ALS=0.                                                            HSL 0046
      CLS=0.                                                            HSL 0047
      ZLS=0.                                                            HSL 0048
      BLC=0.                                                            HSL 0049
      BLS=0.                                                            HSL 0050
      A2IN=0.                                                           HSL 0051
      B2IN=0.                                                           HSL 0052
C  RADIAL INTEGRALS                                                     HSL 0053
      DO 20 I=1,NETA                                                    HSL 0054
      ETA(I)=(ETG(I)-EPS)                                               HSL 0055
      EDETA=ETA(I)*DETA(I)                                              HSL 0056
      GDETA=ETG(I)*DETA(I)                                              HSL 0057
      ALZ=ALZ+EDETA*AMBO(I)                                             HSL 0058
      CLZ=CLZ+GDETA*AMBO(I)                                             HSL 0059
      ZLZ=ZLZ+ETA(I)*GDETA*AMBO(I)                                      HSL 0060
      ALC=ALC+EDETA*CLAM(2,I)                                           HSL 0061
      ALS=ALS+ETG(I)*EDETA*SLAM(1,I)                                    HSL 0062
      CLS=CLS+DETA(I)*SLAM(1,I)                                         HSL 0063
      ZLS=ZLS+EDETA*SLAM(1,I)                                           HSL 0064
      BLC=BLC+EDETA*ETG(I)*CLAM(1,I)                                    HSL 0065
      BLS=BLS+EDETA*SLAM(2,I)                                           HSL 0066
      A2IN=A2IN+EDETA*(HEMU*(SLAM(1,I)-SLAM(3,I))-ETG(I)*CLAM(2,I))     HSL 0067
      B2IN=B2IN+EDETA*(HEMU*(CLAM(1,I)-CLAM(3,I))+ETG(I)*SLAM(2,I))     HSL 0068
20    CONTINUE                                                          HSL 0069
C  COLLECTIVE PITCH                                                     HSL 0070
      THAF=6.0*EMU*ENUM/DENT                                            HSL 0071
      THCT=3.0*CTSA/DENTH                                               HSL 0072
```

```
      THTW=TH1*(-0.75*(OE4+OE2*EMS)+12.0*EMS*ENUM*EIN2/DENT)/DENTH       HSL 0073
      THLZ=(3.0*CLZ-THAF*EMU*ALZ)/DENTH                                  HSL 0074
      THLC=THAF*HEMU*ALC/DENTH                                           HSL 0075
      THLS=(1.5*EMU*CLS-THAF*ALS)/DENTH                                  HSL 0076
      THETA0=THCT+THTW+THLZ+THLC+THLS                                    HSL 0077
C   NO FEATHERING PLANE FLAPPING HARMONICS                               HSL 0078
      A0AF=4.0/DENT                                                      HSL 0079
      A0TF=8.0*EMU*EIN1/DENT                                             HSL 0080
      A0TW=8.0*EMU*EIN2*TH1/DENT+A0TF*THTW                               HSL 0081
      A0CT=A0TF*THCT                                                     HSL 0082
      A0LZ=A0TF*THLZ-A0AF*EMU*ALZ                                        HSL 0083
      A0LC=A0TF*THLC+A0AF*HEMU*ALC                                       HSL 0084
      A0LS=A0TF*THLS-A0AF*ALS                                            HSL 0085
      A0NE=A0TW+A0CT+A0LZ+A0LC+A0LS                                      HSL 0086
      AZTF=HALN*(EIN2+HEMS*EINZ)                                         HSL 0087
      AZAF=HALN*FEPSI*EMU                                                HSL 0088
      AZTW=HALN*(EIN3+HEMS*EIN1)*TH1+AZTF*THTW-AZAF*A0TW                 HSL 0089
      AZCT=AZTF*THCT-AZAF*A0CT                                           HSL 0090
      AZLZ=AZTF*THLZ-AZAF*A0LZ-HALN*ZLZ                                  HSL 0091
      AZLC=AZTF*THLC-AZAF*A0LC                                           HSL 0092
      AZLS=AZTF*THLS-AZAF*A0LS-HALN*HEMU*ZLS                             HSL 0093
      AZERO=AZTW+AZCT+AZLZ+AZLC+AZLS                                     HSL 0094
      B0ZF=4.0*EMU*EIN1/DENB                                             HSL 0095
      B0BF=4.0/DENB                                                      HSL 0096
      B0TW=B0ZF*AZTW                                                     HSL 0097
      B0CT=B0ZF*AZCT                                                     HSL 0098
      B0LZ=B0ZF*AZLZ                                                     HSL 0099
      B0LC0=B0BF*BLC                                                     HSL 0100
      B0LS0=B0ZF*AZLS                                                    HSL 0101
      B0LCT=B0ZF*AZLC                                                    HSL 0102
      B0LST=B0BF*HEMU*BLS                                                HSL 0103
      B0NE=B0TW+B0CT+B0LZ+B0LC0+B0LS0+B0LCT+B0LST                        HSL 0104
      A1MU=EMU*A0NE                                                      HSL 0105
      IF (IGO.EQ.0) GO TO 40                                             HSL 0106
      A2C=A2IN-HEMS*TH1*EIN1                                             HSL 0107
      B2C=-HEMS*AZERO*EINZ+FEPSI*EMU*B0NE-B2IN                           HSL 0108
```

```
      ATWO=FLN*(-HEMS*THETAO*EINZ+FEPSI*A1MU+A2C+EFLN*B2C)/(1.+EFLN**2)        HSL 0109
      BTWO=FLN*B2C-EFLN*ATWO                                                   HSL 0110
C  RECOMPUTE USING 2ND HARMONIC FLAPPING                                       HSL 0111
      THBT=3.0*EMS*BTWO*(-0.25*OEI+ENUM*FTWO/DENT)/DENTH                       HSL 0112
      THETAO=THETAO+THBT                                                       HSL 0113
      AOBT=AOTF*THBT+2.0*EMU*FTWO*BTWO/DENT                                    HSL 0114
      AONE=AONE+AOBT                                                           HSL 0115
      AZBT=AZTF*THBT-AZAF*AOBT+0.5*HALN*HEMS*EINZ*BTWO                         HSL 0116
      AZERO=AZERO+AZBT                                                         HSL 0117
      BOBT=BOZF*AZBT                                                           HSL 0118
      BOAT=-2.0*EMU*FTWO*ATWO/DENB                                             HSL 0119
      BONE=BONE+BOBT+BOAT                                                      HSL 0120
      A1MU=EMU*AONE                                                            HSL 0121
      B2C=-HEMS*AZERO*EINZ+FEPSI*EMU*BONE-B2IN                                 HSL 0122
      ATWO=FLN*(-HEMS*THETAO*EINZ+FEPSI*A1MU+A2C+EFLN*B2C)/(1.+EFLN**2)        HSL 0123
      BTWO=FLN*B2C-EFLN*ATWO                                                   HSL 0124
C  END HARMONIC FLAPPING COMPUTATION                                           HSL 0125
30    CONTINUE                                                                 HSL 0126
      PRINT 101,     AZERO,AONE,BONE,ATWO,BTWO,THETAO                          HSL 0127
101   FORMAT (1H0/'0HARMONICS  OF  FLAPPING'/'0AZERO=',F8.5/'0AONE=',          HSL 0128
     1 F8.5,'  BONE=',F8.5/'0ATWO=',F8.5,'  BTWO=',F8.5/'0THETAO=',F8.5)       HSL 0129
      PRINT 102,        THETAO,THCT,THTW,THLZ,THLC,THLS,THBT,                  HSL 0130
     1 AZERO,AZCT,AZTW,AZLZ,AZLC,AZLS,AZBT,                                    HSL 0131
     2 AONE,AOCT,AOTW,AOLZ,AOLC,AOLS,AOBT,                                     HSL 0132
     3 BONE,BOCT,BOTW,BOLZ,BOLCT,BOLSO,BOBT,BOAT,BOLCO,BOLST                   HSL 0133
102   FORMAT (//'0    FUNCTION      = F(CT)    + F(TH1)   + F(L-0)   + F(L-    HSL 0134
     12C) + F(L-1S) + F(B2)    + F(A2)    + F(L-1C) + F(L-2S)'/                HSL 0135
     2 '0THETAO =',F8.5,' =',5(F8.5,2X),F8.5/                                  HSL 0136
     3 '0AZERO  =',F8.5,' =',5(F8.5,2X),F8.5/                                  HSL 0137
     4 '0AONE   =',F8.5,' =',5(F8.5,2X),F8.5/                                  HSL 0138
     5 '0BONE   =',F8.5,' =',8(F8.5,2X),F8.5)                                  HSL 0139
      PRINT 103,        ALZ,CLZ,ZLZ,BLC,CLS,ZLS,ALS,ALC,BLS                    HSL 0140
103   FORMAT (//'0LAMBDA  INTEGRALS - ETA=(ETG-EPS)/(1.0-EPS)'/                HSL 0141
     1 '0ETA*L-0=',F9.6,'  ETG*L-0=',F9.6,'  ETA*ETG*L-0=',F9.6,               HSL 0142
     2 '  ETA*ETG*L-1C=',F9.6/                                                 HSL 0143
     3 '0L-1S=',F9.6,'  ETA*L-1S=',F9.6,'  ETA*ETG*L-1S=',F9.6,                HSL 0144
```

```
     4 '   ETA*L-2C=',F9.6,'   ETA*L-2S=',F9.6)                                                  HSL 0145
       RETURN                                                                                 HSL 0146
40     ATWO=0.0                                                                               HSL 0147
       BTWO=0.0                                                                               HSL 0148
       GO TO 30                                                                               HSL 0149
       END                                                                                    HSL 0150
```

```
      SUBROUTINE BLADE                                                  BLDE0001
C  LDS-73                                                               BLDE0002
C  COMPUTE BLADE MOTION DYNAMICS INCLUDING FLAPPING AND 1ST MODE BENDING BLDE0003
      REAL K,L,LN,M1,M2,M4,M5,M6,M7                                     BLDE0004
      DIMENSION ALP(6,49),Q1(150),QD1(150),Q2(150),QD2(150),            BLDE0005
     1 G(6),GP(6),TH(6),NQ(20),QC1(9),QS1(9),QC2(9),QS2(9),             BLDE0006
     2 QA1(9),QP1(9),QA2(9),QP2(9),                                     BLDE0007
     3 AIRL(6,25),ALPHS(6,25),DZDR(6,25)                                BLDE0008
      EQUIVALENCE (SUM(1),ALP(1)),(SUM(300),Q1(1)),(SUM(451),QD1(1)),   BLDE0009
     1 (SUM(601),Q2(1)),(SUM(751),QD2(1)),(SUM(1051),AIRL(1)),          BLDE0010
     2 (SUM(1201),ALPHS(1)),(SUM(1651),DZDR(1))                         BLDE0011
      COMMON/SINCO/S(97),C(97),UTS(6,25)                                BLDE0012
      COMMON/SGAM/NETA,NPSIO,NPSO,NG,   WF,EMU,A,BOER,DANGO,            BLDE0013
     1 SUM(6,25,18),ETP(6),DETA(6),ANG(25),PSI(25),ALPH(6,25)           BLDE0014
      COMMON/SAMB/NH,AMBDA(25,6),AMBO(6),CLAM(9,6),SLAM(9,6)            BLDE0015
      COMMON/SLDS/IRB,IRL,IPL,IPRF,IUT,NI,                              BLDE0016
     1 CTO,SM,TH1,VT,EPS,LN,F,CON,ALPS,COSI,GMA,ELO,OMEPS,              BLDE0017
     2 EMT,EMS,DENTH,A1MU,AZERO,AONE,BONE,ATWO,BTWO,THETAO,             BLDE0018
     3 ETA(6),TC(6),ELDS(6),TW(6),GMC(6,25)                             BLDE0019
      NAMELIST/BEND/M1,M2,M4,M5,M6,M7,K,NQIM,NQCM,NTIM,EQ,ECT,IPRQ      BLDE0020
      IF (IRB.LT.2) READ (5,BEND)                                       BLDE0021
      OMEPS=1.-EPS                                                      BLDE0022
      THETA=THETAO                                                      BLDE0023
      NPSI=NPSIO                                                        BLDE0024
      DANG=DANGO                                                        BLDE0025
      WFO=1.0-WF                                                        BLDE0026
      SMAT=.5*SM*A                                                      BLDE0027
      CTSA=(1.0-F)*GMA*CTO/SMAT                                         BLDE0028
      DTHD=DENTH*SMAT/(3.0*GMA)                                         BLDE0029
      EMK1=M1+K                                                         BLDE0030
      EMK2=M2-3.*K                                                      BLDE0031
      EMK4=M4+M7+9.*K                                                   BLDE0032
      NPH=1                                                             BLDE0033
      NQC=0                                                             BLDE0034
      NTI=0                                                             BLDE0035
      DO 10 I=1,NETA                                                    BLDE0036
```

```
      ETA(I)=(ETP(I)-EPS)/OMEPS
C  FLAPWISE BENDING MODE SHAPE
      G(I)=ETA(I)*(4.*ETA(I)-3.)
C  RADIAL DERIVATIVE OF G
      GP(I)=8.*ETA(I)-3.
C  TWISTED BLADE PITCH DISTRIBUTION
10    TH(I)=THETA0+TW(I)
11    NQI=0
      DPS=DANG*DANG
      DPST=2./DPS
      SDP=6.*DANG
      SDPE=11./SDP
      DO 15 J=1,NPSI
      CTP=2.*C(J)*C(J)-1.
      STP=2.*C(J)*S(J)
C  USE HARMONIC SOLUTION FLAPPING FOR INITIAL ESTIMATE OF Q1
      Q1(J)=AZERO-AONE*C(J)-ATWO*CTP-BONE*S(J)-BTWO*STP
      Q2(J)=0.
      QD1(J)=AONE*S(J)-BONE*C(J)+2.*(ATWO*STP-BTWO*CTP)
15    QD2(J)=0.
20    J1=NPSI-1
      J2=NPSI-2
      J3=NPSI-3
      IF (NPSI.GT.49) GO TO 209
C  STORE PREVIOUS SOLUTIONS
      JB=NPSI+1
      IF (NQI.EQ.0) JB=1
204   JE=JB+NPSI-1
      DO 205 J=JB,JE
      JS=J+NPSI
      Q1(JS)=Q1(J)
      Q2(JS)=Q2(J)
      QD1(JS)=QD1(J)
205   QD2(JS)=QD2(J)
      IF (JB.EQ.1) GO TO 206
      JB=1
```

```
      GO TO 204                                                         BLDE0073
  206 IF (NQI.LT.5) GO TO 209                                           BLDE0074
C  AVERAGE LAST TWO SOLUTIONS                                           BLDE0075
      DO 208 J=1,NPSI                                                   BLDE0076
      JS=J+NPSI                                                         BLDE0077
      JO=JS+NPSI                                                        BLDE0078
      Q1(J)=WF*Q1(JS)+WFO*Q1(JO)                                        BLDE0079
      Q2(J)=WF*Q2(JS)+WFO*Q2(JO)                                        BLDE0080
      QD1(J)=WF*QD1(JS)+WFO*QD1(JO)                                     BLDE0081
  208 QD2(J)=WF*QD2(JS)+WFO*QD2(JO)                                     BLDE0082
C  SET 0 AND 360 EQUAL                                                  BLDE0083
  209 Q1(1)=Q1(NPSI)                                                    BLDE0084
      Q2(1)=Q2(NPSI)                                                    BLDE0085
      QD1(1)=QD1(NPSI)                                                  BLDE0086
      QD2(1)=QD2(NPSI)                                                  BLDE0087
C  START PSI LOOP                                                       BLDE0088
      DO 30 J=1,NPSI                                                    BLDE0089
      EMC=EMU*C(J)                                                      BLDE0090
C  INITIALIZE CONSTANTS                                                 BLDE0091
      C1=0.                                                             BLDE0092
      C2=0.                                                             BLDE0093
      C3=0.                                                             BLDE0094
      C4=0.                                                             BLDE0095
      C5=0.                                                             BLDE0096
      C6=0.                                                             BLDE0097
      C8=0.                                                             BLDE0098
      C9=0.                                                             BLDE0099
      C10=0.                                                            BLDE0100
C  DEFINE INTERPOLATION PARAMETERS                                      BLDE0101
      JL=1+(J-1)/NPH                                                    BLDE0102
      JU=JL+1                                                           BLDE0103
      IF (JU.GT.NPSIO) JU=2                                             BLDE0104
      DAMB=FLOAT(J-1-NPH*(JL-1))/FLOAT(NPH)                             BLDE0105
C  START ETA LOOP                                                       BLDE0106
      DO 28 I=1,NETA                                                    BLDE0107
C  INTERPOLATE GLAUERT MACH NO. CORRECTION                              BLDE0108
```

```
      GM=GMC(I,JL)+DAMB*(GMC(I,JU)-GMC(I,JL))                           BLDE0109
      GMEPS=GM*OMEPS**2                                                 BLDE0110
C  INTERPOLATE LAMBDA (DOWNWASH) AND CONVERT FROM LAMBDA-TPP            BLDE0111
C  TO LAMBDA-NFP BY ADDING A1MU                                         BLDE0112
      AMB=AMBDA(JL,I)+DAMB*(AMBDA(JU,I)-AMBDA(JL,I))+A1MU               BLDE0113
C  INTERPOLATE UT                                                       BLDE0114
      UT=UTS(I,JL)+DAMB*(UTS(I,JU)-UTS(I,JL))                           BLDE0115
      THT=UT*TH(I)                                                      BLDE0116
      UP=AMB+(Q1(J)+Q2(J)*GP(I))*EMC+ETA(I)*QD1(J)+G(I)*QD2(J)          BLDE0117
C  ANGLE OF ATTACK                                                      BLDE0118
      ALP(I,J)=TH(I)-ATAN(UP/UT)                                        BLDE0119
C  REVERSE FLOW                                                         BLDE0120
      IF (UT.LE.0.0) ALP(I,J)=-ALP(I,J)                                 BLDE0121
C  STALL TEST                                                           BLDE0122
      IF (ABS(ALP(I,J)).GT.ALPS) GO TO 22                               BLDE0123
21    UTD=ABS(UT)*DETA(I)*F/GMEPS                                       BLDE0124
      UTDE=UTD*ETA(I)                                                   BLDE0125
      UTDG=UTD*G(I)                                                     BLDE0126
      EMCGP=EMC*GP(I)                                                   BLDE0127
      C1=C1-(AMB-THT)*UTDE+CTSA*DETA(I)*ETA(I)/OMEPS**2                 BLDE0128
      C2=C2+UTDE*ETA(I)                                                 BLDE0129
      C3=C3+UTDE*G(I)                                                   BLDE0130
      C4=C4+UTDE*EMC                                                    BLDE0131
      C5=C5+EMCGP*UTDE                                                  BLDE0132
      C6=C6-(AMB-THT)*UTDG+CTSA*DETA(I)*G(I)/OMEPS**2                   BLDE0133
      C8=C8+UTDG*G(I)                                                   BLDE0134
      C9=C9+UTDG*EMC                                                    BLDE0135
      C10=C10+EMCGP*UTDG                                                BLDE0136
      GO TO 28                                                          BLDE0137
C  STALLED CASE                                                         BLDE0138
22    UTAD=SIGN(F*ALPS*DETA(I)*UT**2/GMEPS,ALP(I,J))                    BLDE0139
      C1=C1+UTAD*ETA(I)+CTSA*DETA(I)*ETA(I)/OMEPS**2                    BLDE0140
      C6=C6+UTAD*G(I)+CTSA*DETA(I)*G(I)/OMEPS**2                        BLDE0141
C  END ETA LOOP                                                         BLDE0142
28    CONTINUE                                                          BLDE0143
      A1=DPST+(SDPE*C2+EMK1+C4)/M5                                      BLDE0144
```

```
       B1=(SDPE*C3+EMK2+C5)/M5                                                  BLDE0145
       A2=(SDPE*C3+EMK2+C9)/M6                                                  BLDE0146
       B2=DPST+(SDPE*C8+EMK4+C10)/M6                                            BLDE0147
       QD0=18.*Q1(J1)-9.*Q1(J2)+2.*Q1(J3)                                       BLDE0148
       QDT=18.*Q2(J1)-9.*Q2(J2)+2.*Q2(J3)                                       BLDE0149
       Y1=(5.*Q1(J1)-4.*Q1(J2)+Q1(J3))/DPS+(C1+(C2*QD0+C3*QDT)/SDP)/M5          BLDE0150
       Y2=(5.*Q2(J1)-4.*Q2(J2)+Q2(J3))/DPS+(C6+(C3*QD0+C8*QDT)/SDP)/M6          BLDE0151
       DEN=A1*B2-B1*A2                                                          BLDE0152
C  FLAPPING AMPLITUDE                                                           BLDE0153
       Q1(J)=(Y1*B2-B1*Y2)/DEN                                                  BLDE0154
C  BENDING AMPLITUDE                                                            BLDE0155
       Q2(J)=(A1*Y2-Y1*A2)/DEN                                                  BLDE0156
C  DERIVATIVES WITH RESPECT TO TIME (AZIMUTH)                                   BLDE0157
       QD1(J)=(11.*Q1(J)-QD0)/SDP                                               BLDE0158
       QD2(J)=(11.*Q2(J)-QDT)/SDP                                               BLDE0159
       J3=J2                                                                    BLDE0160
       J2=J1                                                                    BLDE0161
       J1=J                                                                     BLDE0162
C  END PSI LOOP                                                                 BLDE0163
30     CONTINUE                                                                 BLDE0164
       IQT=1                                                                    BLDE0165
       NQI=NQI+1                                                                BLDE0166
C  CONVERGENCE TESTS                                                            BLDE0167
       IF (ABS(Q1(1)-Q1(NPSI)).GT.EQ) GO TO 32                                  BLDE0168
       IF (ABS(Q2(1)-Q2(NPSI)).GT.EQ) GO TO 32                                  BLDE0169
       IF (ABS(QD1(1)-QD1(NPSI)).GT.EQ) GO TO 32                                BLDE0170
       IF (ABS(QD2(1)-QD2(NPSI)).LE.EQ) GO TO 40                                BLDE0171
32     IF (NQI.GT.NQIM) GO TO 34                                                BLDE0172
       IF (Q1(J).GT.1.0) GO TO 34                                               BLDE0173
C  COMPUTE HARMONICS OF FLAPPING                                                BLDE0174
       CALL HARO (NH,NPSI,DANG,Q1,QZ1,QC1,QS1)                                  BLDE0175
       A1MU=-EMU*QC1(1)                                                         BLDE0176
       GO TO 20                                                                 BLDE0177
C  EITHER DID NOT CONVERGE IN NQIM ITERATIONS OR BLEW UP                        BLDE0178
C  TRY HALVING AZIMUTH INTERVAL SIZE                                            BLDE0179
C  IF MORE THAN NQCM HALVINGS NEEDED - GIVE UP                                  BLDE0180
```

```
34       PRINT 341,        NPH,NQI,NTI,THETA,CT,CTO                               BLDE0181
341      FORMAT ('1NPH=',I1,' NQI=',I3,' NTI=',I2,' THETA=',F7.5,                 BLDE0182
     1  ' CT=',F8.6,' CTO=',F8.6/                                                 BLDE0183
     2  'OBLADE MOTION FROM N, N-1, AND N-2 ITERATIONS'/                          BLDE0184
     3  '0 J',3('      Q1    '),3('      Q2    '),3('     QD1    '),              BLDE0185
     4  3('    QD2    '))                                                         BLDE0186
         DO 343 J=1,NPSI                                                          BLDE0187
         JS=J+NPSI                                                                BLDE0188
         JO=JS+NPSI                                                               BLDE0189
343      PRINT 344,        J,Q1(J),Q1(JS),Q1(JO),Q2(J),Q2(JS),Q2(JO),             BLDE0190
     1  QD1(J),QD1(JS),QD1(JO),QD2(J),QD2(JS),QD2(JO)                             BLDE0191
344      FORMAT (1X,I2,12(1X,F9.6))                                               BLDE0192
         IF (IQT.EQ.2) GO TO 47                                                   BLDE0193
         NQC=NQC+1                                                                BLDE0194
         IF (NQC.GT.NQCM) GO TO 40                                                BLDE0195
         NPSI=2*(NPSI-1)+1                                                        BLDE0196
         DANG=DANG/2.                                                             BLDE0197
         NPH=2*NPH                                                                BLDE0198
         CD=COS(DANG)                                                             BLDE0199
         SD=SIN(DANG)                                                             BLDE0200
         DO 35 J=2,NPSI                                                           BLDE0201
         JG=J-1                                                                   BLDE0202
         C(J)=CD*C(JG)-SD*S(JG)                                                   BLDE0203
35       S(J)=CD*S(JG)+SD*C(JG)                                                   BLDE0204
         A1MU=EMU*AONE                                                            BLDE0205
         DO 36 I=1,NETA                                                           BLDE0206
36       TH(I)=THETA0+TW(I)                                                       BLDE0207
         GO TO 11                                                                 BLDE0208
C   CONVERGED BLADE MOTION SOLUTION                                               BLDE0209
C   COMPUTE HARMONICS OF FLAPPING                                                 BLDE0210
40       CALL HARO (NH,NPSI,DANG,Q1,QZ1,QC1,QS1)                                  BLDE0211
         A1MU=-EMU*QC1(1)                                                         BLDE0212
         CT=(1.-F)*CTO                                                            BLDE0213
         NPS=NPSI-1                                                               BLDE0214
         SMATS=F*SMAT/FLOAT(NPS)                                                  BLDE0215
C   COMPUTE CT (THRUST COEFFICIENT)                                               BLDE0216
```

```
      DO 45 J=1,NPS                                                     BLDE0217
      JL=1+(J-1)/NPH                                                    BLDE0218
      JU=JL+1                                                           BLDE0219
      IF (JU.GT.NPSIO) JU=2                                             BLDE0220
      DAMB=FLOAT(J-1-NPH*(JL-1))/FLOAT(NPH)                             BLDE0221
      DO 44 I=1,NETA                                                    BLDE0222
      GM=GMC(I,JL)+DAMB*(GMC(I,JU)-GMC(I,JL))                           BLDE0223
      UT=UTS(I,JL)+DAMB*(UTS(I,JU)-UTS(I,JL))                           BLDE0224
      ALC=ALP(I,J)                                                      BLDE0225
      IF (ABS(ALC).GT.ALPS) ALC=SIGN(ALPS,ALC)                          BLDE0226
 44   CT=CT+SMATS*UT*UT*ALC*DETA(I)/GM                                  BLDE0227
 45   CONTINUE                                                          BLDE0228
C  TEST FOR CT CONVERGENCE                                              BLDE0229
      IF (ABS(CT-CTO).LE.ECT) GO TO 48                                  BLDE0230
C  ADJUST COLLECTIVE PITCH TO GET CORRECT CT                            BLDE0231
      NTI=NTI+1                                                         BLDE0232
      NO(NTI)=NQI                                                       BLDE0233
      IQT=2                                                             BLDE0234
      IF (NTI.GT.NTIM) GO TO 34                                         BLDE0235
      DTH=(CTO-CT)/DTHD                                                 BLDE0236
      THETA=THETA+DTH                                                   BLDE0237
      DO 46 I=1,NETA                                                    BLDE0238
 46   TH(I)=TH(I)+DTH                                                   BLDE0239
      NQI=0                                                             BLDE0240
      GO TO 20                                                          BLDE0241
C  IF MORE THAN NTIM PITCH ADJUSTMENTS NEEDED - GIVE UP                 BLDE0242
 47   CONTINUE                                                          BLDE0243
C  CT CONVERGENCE                                                       BLDE0244
 48   CONTINUE                                                          BLDE0245
C  COMPUTE HARMONICS OF FLAPWISE BENDING                                BLDE0246
      CALL HARD (NH,NPSI,DANG,Q2,QZ2,QC2,QS2)                           BLDE0247
      PRINT 100,       THETA, M1,M5,K,                                  BLDE0248
     1 EO,M2,M6,NQI,                                                    BLDE0249
     2 ECT,M4,M7,NTI                                                    BLDE0250
100   FORMAT ('1BLADE  BENDING  SOLUTION'/                              BLDE0251
     1 '0THETA=',F7.4,'   M1=',F7.4,'  M5=',F7.4,'  K=',F7.4/           BLDE0252
```

```
     2 'OEQ=',F7.6,'      M2=',F7.4,'  M6=',F7.4,'  NQI=',I3/            BLDE0253
     3 'OECT=',F7.6,'     M4=',F7.4,'  M7=',F7.4,'  NTI=',I3)            BLDE0254
C  COMPUTE MAGNITUDE AND PHASE OF BLADE MOTION HARMONICS                 BLDE0255
      DO 481 N=1,NH                                                      BLDE0256
      QA1(N)=SQRT(QC1(N)**2+QS1(N)**2)                                   BLDE0257
      QA2(N)=SQRT(QC2(N)**2+QS2(N)**2)                                   BLDE0258
      QP1(N)=57.29578*ATAN2(QC1(N),QS1(N))                               BLDE0259
481   QP2(N)=57.29578*ATAN2(QC2(N),QS2(N))                               BLDE0260
      PRINT 482                                                          BLDE0261
482   FORMAT ('OHARMONICS OF BLADE MOTION'/                              BLDE0262
     1 'O',14X,'Q1',19X,'Q2',21X,'Q1',17X,'Q2'/                          BLDE0263
     2 'ONH ',2(6X,'COS',7X,'SIN  '),2X,2('  AMPLITUDE   PHASE'))        BLDE0264
      PRINT 483,       QZ1,QZ2,QZ1,QZ2                                   BLDE0265
483   FORMAT ('0 0 ',F11.5,10X,F11.5,12X,F11.5,8X,F11.5)                 BLDE0266
      DO 484 N=1,NH                                                      BLDE0267
484   PRINT 485,       N, QC1(N),QS1(N),QC2(N),QS2(N),QA1(N),QP1(N),     BLDE0268
     1 QA2(N),QP2(N)                                                     BLDE0269
485   FORMAT (1X,I2,1X,2(F11.5,F10.5),2X,2(F11.5,F8.1))                  BLDE0270
      AZERO=QZ1                                                          BLDE0271
      AONE=-QC1(1)                                                       BLDE0272
      BONE=-QS1(1)                                                       BLDE0273
      ATWO=-QC1(2)                                                       BLDE0274
      BTWO=-QS1(2)                                                       BLDE0275
      DO 50 M=1,NPSIO                                                    BLDE0276
      J=1+NPH*(M-1)                                                      BLDE0277
      C(M)=C(J)                                                          BLDE0278
      S(M)=S(J)                                                          BLDE0279
      Q1(M)=Q1(J)                                                        BLDE0280
      Q2(M)=Q2(J)                                                        BLDE0281
      QD1(M)=QD1(J)                                                      BLDE0282
      QD2(M)=QD2(J)                                                      BLDE0283
      DO 49 I=1,NETA                                                     BLDE0284
49    ALPH(I,M)=ALP(I,J)                                                 BLDE0285
50    CONTINUE                                                           BLDE0286
      PRINT 101                                                          BLDE0287
101   FORMAT ('OPSI',5X,'Q1',9X,'QD1',8X,'Q2',9X,'QD2',8X,'COS',8X,      BLDE0288
```

```
     1 'SIN')                                                           BLDE0289
      DO 55 I=1,NETA                                                    BLDE0290
      ELDS(I)=0.0                                                       BLDE0291
      DO 54 J=1,NPSIO                                                   BLDE0292
      UT=UTS(I,J)                                                       BLDE0293
      ALC=ALPH(I,J)                                                     BLDE0294
      IF (ABS(ALC).GT.ALPS) ALC=SIGN(ALPS,ALC)                          BLDE0295
      AIRL(I,J)=UT*UT*ALC                                               BLDE0296
      ALPHS(I,J)=ALC                                                    BLDE0297
54    ELDS(I)=ELDS(I)+AIRL(I,J)                                         BLDE0298
55    ELDS(I)=ELDS(I)-AIRL(I,J)                                         BLDE0299
      OFNP=(1.-F)/(NPSIO-1)                                             BLDE0300
      DO 56 I=1,NETA                                                    BLDE0301
      DLDS=OFNP*ELDS(I)                                                 BLDE0302
      DO 56 J=1,NPSIO                                                   BLDE0303
      AIRL(I,J)=F*AIRL(I,J)+DLDS                                        BLDE0304
C  DELTA Z/DELTA R RELATIVE TO TPP                                      BLDE0305
      DZDR(I,J)=Q1(J)+GP(I)*Q2(J)+AONE*C(J)+BONE*S(J)                   BLDE0306
56    CONTINUE                                                          BLDE0307
      DO 57 J=1,NPSIO                                                   BLDE0308
57    PRINT 102,        ANG(J),Q1(J),QD1(J),Q2(J),QD2(J),C(J),S(J)      BLDE0309
102   FORMAT (1H ,F4.0,6F11.7)                                          BLDE0310
      PRINT 103,        (NQ(I),I=1,NTI)                                 BLDE0311
103   FORMAT ('0NQI=',20I5)                                             BLDE0312
      IF (IPRQ.GT.0.AND.IPRQ.GT.NG) RETURN                              BLDE0313
C  OUTPUT COMPONENTS OF LOADS                                           BLDE0314
      DO 60 I=1,NETA                                                    BLDE0315
      ODD=FLOAT(I)/2.0-I/2                                              BLDE0316
      IF (ODD.GT.0.1) PRINT 105                                         BLDE0317
      PRINT 106,ETP(I)                                                  BLDE0318
      DO 60 J=1,NPSIO                                                   BLDE0319
      UT=UTS(I,J)                                                       BLDE0320
      QO=Q1(J)*EMU*C(J)                                                 BLDE0321
      QT=Q2(J)*GP(I)*EMU*C(J)                                           BLDE0322
      QDO=ETA(I)*QD1(J)                                                 BLDE0323
      QDT=G(I)*QD2(J)                                                   BLDE0324
```

```
      UP=AMBDA(J,I)+A1MU+Q0+QT+QD0+QDT                                  BLDE0325
      UPT=UP/UT                                                         BLDE0326
      PRINT 107,       ANG(J),AIRL(I,J),ALPH(I,J),TH(I),UPT,UT,UP,      BLDE0327
     1 AMBDA(J,I),A1MU,Q0,QT,QD0,QDT,DZDR(I,J)                          BLDE0328
   60 CONTINUE                                                          BLDE0329
  105 FORMAT ('1COMPONENTS OF LOADS')                                   BLDEC330
  106 FORMAT ('0ETA=',F5.3/                                             BLDE0331
     1 ' PSI     LOADS     ALPHA     THETA     UP/UT     UT        UP       LAM BLDE0332
     2BDA    MU*A1     Q1        Q2        QD1       QD2       DZDR')   BLDE0333
  107 FORMAT (1X,F4.0,13F9.5)                                           BLDE0334
      RETURN                                                            BLDE0335
      END                                                               BLDE0336
```

```
      SUBROUTINE HARO(NH,NPSI,DANG,T,AZ,A,B)                            HAR00001
C  LDS-73                                                               HAR00002
C  HARMONIC ANALYSIS OF 1-D VARIABLE T(PSI) FOR PSI = 0 - 360           HAR00003
      REAL T(97),A(9),B(9)                                              HAR00004
      Q=DANG/9.424778                                                   HAR00005
      NPS=NPSI-1                                                        HAR00006
      DO 10 N=1,NH                                                      HAR00007
      DPSIN=DANG*FLOAT(N)                                               HAR00008
      CD=COS(DPSIN)                                                     HAR00009
      SD=SIN(DPSIN)                                                     HAR00010
      C=CD                                                              HAR00011
      S=SD                                                              HAR00012
      A(N)=T(1)+T(NPSI)+4.0*CD*T(2)                                     HAR00013
      B(N)=4.0*SD*T(2)                                                  HAR00014
      DO 8 J=3,NPS,2                                                    HAR00015
      CA=CD*C-SD*S                                                      HAR00016
      SA=CD*S+SD*C                                                      HAR00017
      C=CD*CA-SD*SA                                                     HAR00018
      S=CD*SA+SD*CA                                                     HAR00019
      A(N)=A(N)+2.0*CA*T(J)+4.0*C*T(J+1)                                HAR00020
    8 B(N)=B(N)+2.0*SA*T(J)+4.0*S*T(J+1)                                HAR00021
      A(N)=Q*A(N)                                                       HAR00022
   10 B(N)=Q*B(N)                                                       HAR00023
      AZ=T(1)+T(NPSI)+4.0*T(2)                                          HAR00024
      DO 12 J=3,NPS,2                                                   HAR00025
   12 AZ=AZ+2.0*T(J)+4.0*T(J+1)                                         HAR00026
      AZ=0.5*Q*AZ                                                       HAR00027
      RETURN                                                            HAR00028
      END                                                               HAR00029
```

```
      SUBROUTINE HART                                                   HART0001
C  LDS-73                                                               HART0002
C  HARMONIC ANALYSIS OF AMB(PSI,ETA) FOR PSI=0-360                      HART0003
      COMMON/SGAM/NE,  NPSI, NPS,NG,   WF,EMU,A,BOER,DANG,              HART0004
     1 SUM(6,25,18),ETG(6),DETA(6),ANG(25),PSI(25),ALPHA(6,25)          HART0005
      COMMON/SAMB/NH,AMB(25,6),  AMBO(6),CLAM(9,6),SLAM(9,6)            HART0006
      Q=DANG/9.424778                                                   HART0007
      NPS=NPSI-1                                                        HART0008
      DO 10 N=1,NH                                                      HART0009
      DPSIN=DANG*FLOAT(N)                                               HART0010
      CD=COS(DPSIN)                                                     HART0011
      SD=SIN(DPSIN)                                                     HART0012
      C=CD                                                              HART0013
      S=SD                                                              HART0014
      DO 4 K=1,NE                                                       HART0015
      CLAM(N,K)=AMB(1,K)+AMB(NPSI,K)+4.0*CD*AMB(2,K)                    HART0016
4     SLAM(N,K)=4.0*SD*AMB(2,K)                                         HART0017
      DO 8 J=3,NPS,2                                                    HART0018
      CA=CD*C-SD*S                                                      HART0019
      SA=CD*S+SD*C                                                      HART0020
      C=CD*CA-SD*SA                                                     HART0021
      S=CD*SA+SD*CA                                                     HART0022
      DO 8 K=1,NE                                                       HART0023
      CLAM(N,K)=CLAM(N,K)+2.0*CA*AMB(J,K)+4.0*C*AMB(J+1,K)              HART0024
      SLAM(N,K)=SLAM(N,K)+2.0*SA*AMB(J,K)+4.0*S*AMB(J+1,K)              HART0025
8     CONTINUE                                                          HART0026
      DO 10 K=1,NE                                                      HART0027
      CLAM(N,K)=Q*CLAM(N,K)                                             HART0028
10    SLAM(N,K)=Q*SLAM(N,K)                                             HART0029
      DO 14 K=1,NE                                                      HART0030
      AMBO(K)=AMB(1,K)+AMB(NPSI,K)+4.0*AMB(2,K)                         HART0031
      DO 12 J=3,NPS,2                                                   HART0032
12    AMBO(K)=AMBO(K)+2.0*AMB(J,K)+4.0*AMB(J+1,K)                       HART0033
14    AMBO(K)=0.5*Q*AMBO(K)                                             HART0034
      RETURN                                                            HART0035
      END                                                               HART0036
```

```
      SUBROUTINE HALDS
C  LDS-73
C  HARMONIC ANALYSIS OF AIRLOADS
      DIMENSION ELAMP(9,6),ELPH(9,6),AIRL(6,25)
      EQUIVALENCE (SUM(1051),AIRL(1))
      COMMON/SGAM/NETA,NPSI, NPS,NG,   WF,EMU,A,BOER,DANG,
     1 SUM(6,25,18),ETG(6),DETA(6),ANG(25),PSI(25),ALPH(6,25)
      COMMON/SAMB/NH,AMBDA(25,6),AMBO(6),CLAM(9,6),SLAM(9,6)
      COMMON/PLT/IPLX,DEL,ELMN(6),XLDS(6,25)
      IETA=0
      IOUT=IPLX
C  COMPUTED AIRLOADS
      PRINT 950
950   FORMAT ('1COMPUTED  AIRLOADS')
      DO 95 I=1,NETA
      DO 95 J=1,NPSI
      AMBDA(J,I)=AIRL(I,J)
95    CONTINUE
955   CALL HART
C  MAGNITUDE AND PHASE OF AIRLOADS - F(TH)=AMP*SIN(TH+PH)
      DO 96 K=1,NETA
      IF (K.EQ.IETA) GO TO 96
      DO 96 N=1,NH
      ELAMP(N,K)=SQRT(SLAM(N,K)**2+CLAM(N,K)**2)
      ELPH(N,K)=57.29578*ATAN2(CLAM(N,K),SLAM(N,K))
96    CONTINUE
C  PRINT HARMONICS OF AIRLOADS
      PRINT 960,        (ETG(K),K=1,NETA)
960   FORMAT ('0HARMONICS OF AIRLOADS'/'0ETA',9X,F5.3,5(16X,F5.3)/
     1 '0NH ',6(5X,'COS',7X,'SIN',3X)/)
      PRINT 961,        (AMBO(K),K=1,NETA)
961   FORMAT ('   0 ',6(F11.5,10X))
      DO 962 N=1,NH
962   PRINT 963,        N,(CLAM(N,K),SLAM(N,K),K=1,NETA)
963   FORMAT (1X,I2,1X,6(F11.5,F10.5))
      PRINT 964
```

```
964    FORMAT ('0NH ',6(3X,'AMPLITUDE  PHASE  ')/)                             HALD0037
       PRINT 961,       (AMBO(K),K=1,NETA)                                     HALD0038
       DO 965 N=1,NH                                                           HALD0039
965    PRINT 966,       N,(ELAMP(N,K),ELPH(N,K),K=1,NETA)                      HALD0040
966    FORMAT (1X,I2,1X,6(F11.5,F8.1,2X))                                      HALD0041
       IF (IOUT.NE.2) RETURN                                                   HALD0042
       IOUT=0                                                                  HALD0043
C   EXPERIMENTAL AIRLOADS                                                      HALD0044
       PRINT 967                                                               HALD0045
967    FORMAT ('0EXPERIMENTAL  AIRLOADS')                                      HALD0046
       DO 968 I=1,NETA                                                         HALD0047
       IF (XLDS(I,1).EQ.0.0) IETA=I                                            HALD0048
       DO 968 J=1,NPSI                                                         HALD0049
968    AMBDA(J,I)=XLDS(I,J)                                                    HALD0050
       IF (IETA.EQ.0) GO TO 955                                                HALD0051
       DO 969 N=1,NH                                                           HALD0052
       ELAMP(N,IETA)=0.0                                                       HALD0053
969    ELPH(N,IETA)=0.0                                                        HALD0054
       GO TO 955                                                               HALD0055
       END                                                                     HALD0056
```

```
      SUBROUTINE PNTLDS (IPRG)                                          PNTL0001
C  LDS-73                                                               PNTL0002
C  PRINT AIRLOADS, ETC. (UP TO 8 ETA)                                   PNTL0003
      DIMENSION TF(6,25),AF(6,25),HF(6,25),AIRL(6,25),ALPHS(6,25),      PNTL0004
     1 RF(6,25),CTPR(6,25),CDP(6,25),CDMP(6,25),DQP(6,25),DQI(6,25),    PNTLC005
     2 ALPRP(6,25),ALPSS(6,25),EMAP(6,25),EMDP(6,25),DZDR(6,25),        PNTL0006
     3 ELFP(6,25)                                                       PNTL0007
      EQUIVALENCE (SUM(1),TF(1)),(SUM(151),AF(1)),(SUM(301),HF(1)),     PNTL0008
     1 (SUM(451),RF(1)),(SUM(601),CTPR(1)),(SUM(751),CDP(1)),           PNTLC009
     2 (SUM(901),CDMP(1)),(SUM(1051),AIRL(1)),(SUM(1201),ALPHS(1)),     PNTL0010
     3 (SUM(1351),DQP(1)),(SUM(1501),DQI(1)),(SUM(1651),DZDR(1)),       PNTL0011
     4 (SUM(1801),ALPRP(1)),(SUM(1951),ALPSS(1)),(SUM(2101),EMAP(1)),   PNTL0012
     5 (SUM(2251),EMDP(1)),(SUM(2401),ELFP(1))                          PNTL0013
      COMMON/SGAM/NETA,NPSI, NPS,NG,   WF,EMU,A,BOER,DANG,              PNTL0014
     1 SUM(6,25,18),ETG(6),DETA(6),ANG(25),PSI(25),ALPH(6,25)           PNTL0015
      PRINT 106,        (ETG(K),K=1,NETA)                               PNTL0016
      DO 105 J=1,NPSI                                                   PNTL0017
  105 PRINT 107,        ANG(J),(ELFP(K,J),ALPHS(K,J),K=1,NETA)          PNTL0018
  106 FORMAT ('1AIRLOADS(LB/IN)/ALPHA(DEGREES)'/                        PNTL0019
     1 '0ETA=',8(7X,F5.3,4X)/' PSI')                                    PNTL0020
  107 FORMAT (1X,F4.0,8(1X,F7.2,'/',F7.2))                              PNTL0021
      IF (IPRG.GT.0.AND.IPRG.GT.NG) GO TO 126                           PNTL0022
      PRINT 109,        (ETG(K),K=1,NETA)                               PNTL0023
      DO 108 J=1,NPSI                                                   PNTL0024
  108 PRINT 110,        ANG(J),(ALPSS(K,J),ALPRP(K,J),K=1,NETA)         PNTLC025
  109 FORMAT ('0STATIC STALL ALPHA(DEGREES)/REFERENCE ALPHA (DEGREES)'/ PNTL0026
     1 '0ETA=',8(7X,F5.3,4X)/' PSI')                                    PNTL0027
  110 FORMAT (1X,F4.0,8(1X,F7.3,'/',F7.3))                              PNTL0028
      PRINT 112,        (ETG(K),K=1,NETA)                               PNTL0029
      DO 111 J=1,NPSI                                                   PNTL0030
  111 PRINT 107,        ANG(J),(TF(K,J),AF(K,J),K=1,NETA)               PNTLC031
  112 FORMAT ('1THRUST(LB/IN)/AZIMUTHAL FORCE(LB/IN)'/                  PNTL0032
     1 '0ETA=',8(7X,F5.3,4X)/' PSI')                                    PNTL0033
      PRINT 115,        (ETG(K),K=1,NETA)                               PNTL0034
      DO 114 J=1,NPSI                                                   PNTL0035
  114 PRINT 110,        ANG(J),(RF(K,J),HF(K,J),K=1,NETA)               PNTL0036
```

```
115    FORMAT ('ORADIAL FORCE(LB/IN)/HORIZONTAL FORCE(LB/IN)'/                PNTL0037
      1 'OETA=',8(7X,F5.3,4X)/' PSI')                                        PNTL0038
       PRINT 117,       (ETG(K),K=1,NETA)                                    PNTL0039
       DO 116 J=1,NPSI                                                       PNTL0040
116    PRINT 113,       ANG(J),(CDP(K,J),CDMP(K,J),K=1,NETA)                 PNTL0041
117    FORMAT ('1CD/CDM'/                                                    PNTL0042
      1 'OETA=',8(8X,F5.3,3X)/' PSI')                                        PNTL0043
113    FORMAT (1X,F4.0,8(3X,F6.4,'/',F6.4))                                  PNTL0044
       PRINT 119,       (ETG(K),K=1,NETA)                                    PNTL0045
       DO 118 J=1,NPSI                                                       PNTL0046
118    PRINT 113,       ANG(J),(EMAP(K,J),EMDP(K,J),K=1,NETA)                PNTL0047
119    FORMAT ('OBLADE ELEMENT MACH NUMBER/DRAG DIVERGENCE MACH NUMBER'/     PNTL0048
      1 'OETA=',8(8X,F5.3,3X)/' PSI')                                        PNTL0049
       PRINT 121,       (ETG(K),K=1,NETA)                                    PNTL0050
       DO 120 J=1,NPSI                                                       PNTL0051
120    PRINT 107,       ANG(J),(ELFP(K,J),AIRL(K,J),K=1,NETA)                PNTL0052
121    FORMAT ('1DYNAMIC STALL AIRLOADS(LB/IN)/OLD AIRLOADS(LB/IN)'/         PNTL0053
      1 'OETA=',8(7X,F5.3,4X)/' PSI')                                        PNTL0054
       PRINT 123,       (ETG(K),K=1,NETA)                                    PNTL0055
       DO 122 J=1,NPSI                                                       PNTL0056
122    PRINT 107,       ANG(J),(ALPH(K,J),ALPHS(K,J),K=1,NETA)               PNTL0057
123    FORMAT ('OCL/A(DEGREES)/ACTUAL ALPHA(DEGREES)'/                       PNTL0058
      1 'OETA=',8(7X,F5.3,4X)/' PSI')                                        PNTL0059
       PRINT 125,       (ETG(K),K=1,NETA)                                    PNTL0060
       DO 124 J=1,NPSI                                                       PNTL0061
124    PRINT 107,       ANG(J),(ALPHS(K,J),CTPR(K,J),K=1,NETA)               PNTL0062
125    FORMAT ('1ALPHA(DEGREES)/(DELTA CT)*1.0E+6'/                          PNTL0063
      1 'OETA=',8(7X,F5.3,4X)/' PSI')                                        PNTL0064
126    CONTINUE                                                              PNTL0065
       PRINT 128,       (ETG(K),K=1,NETA)                                    PNTL0066
       DO 127 J=1,NPSI                                                       PNTL0067
127    PRINT 110,       ANG(J),(DQP(K,J),DQI(K,J),K=1,NETA)                  PNTL0068
128    FORMAT ('OPARASITE TORQUE/INDUCED TORQUE - MULTIPLY BY 1.0E-6 FOR     PNTL0069
      1 DELTA CQ'/                                                           PNTL0070
      2 'OETA=',8(7X,F5.3,4X)/' PSI')                                        PNTL0071
       RETURN                                                                PNTL0072
```

END

PNTL0073

```
      SUBROUTINE PLTL                                                   PLTL0001
C  LDS-73                                                               PLTL0002
C  CARTESIAN AIRLOADS PLOTS                                             PLTL0003
C  PRINTER VERSION                                                      PLTL0004
C  REQUIRES NPSI.LT.26                                                  PLTL0005
      DIMENSION AIRL(6,25),ELAB(13),ELPLT(128)                          PLTL0006
      EQUIVALENCE (SUM(1051),AIRL(1))                                   PLTL0007
      COMMON/SGAM/NE,  NPSI, NPS,NG,   WF,EMU,A,BUER,DANG,              PLTL0008
     1 SUM(6,25,18),ETA(6),DETA(6),U(25),PSI(25),ALPHA(6,25)            PLTL0009
      COMMON/SQLDS/IABC,IEPS,IS,KZAD,NR,NL,ND,NPHI,NPCP,NPRP,NRNG,      PLTL0010
     1 FTD,RHOK,DPHR,TANMU,AMBDA,                                       PLTL0011
     2 ZETA(6),KPB(25),ELT(25),ELI(25),FTU(25)                          PLTL0012
      COMMON/SLDS/IRB,IRL,IPL,IPRF,IUT,NI,                              PLTL0013
     1 CTO,SM,TH1,VT,EPS,LN,F,CON,ALPS,COSI,GMA,ELO,OMEPS,              PLTL0014
     2 EMT,EMS,DENTH,A1MU,AZERO,AONE,BONE,ATWO,BTWO,THETAO,             PLTL0015
     3 FTE(6),TC(6),ELDS(6),TW(6),GMC(6,25)                             PLTL0016
      COMMON/PLT/IPLX,DEL,ELMN(6),XLDS(6,25)                            PLTL0017
      DATA BLNK,STAR,XPNT/' ','*','X'/                                  PLTL0018
C  TEST FOR NPSI TOO BIG                                                PLTL0019
      IF (NPSI.LT.26) GO TO 10                                          PLTL0020
      PRINT 100,NPSI                                                    PLTL0021
100   FORMAT ('1CALLED PRINTER PLOT FOR NPSI=',I3)                      PLTL0022
      RETURN                                                            PLTL0023
10    IX=1                                                              PLTL0024
C  DEL = AIRLOADS INTERVAL SIZE ON PLOT                                 PLTL0025
C  ELMN = STARTING AIRLOADS VALUE OF PLOT                               PLTL0026
      TDEL=10.0/DEL                                                     PLTL0027
C  START ETA LOOP                                                       PLTL0028
      DO 30 K=1,NE                                                      PLTL0029
C  DEFINE LABEL FOR AIRLOADS AXIS                                       PLTL0030
      ELAB(1)=ELMN(K)                                                   PLTL0031
      DO 12 I=2,13                                                      PLTL0032
12    ELAB(I)=ELAB(I-1)+DEL                                             PLTL0033
      TK=FLOAT(K)/2.0-K/2                                               PLTL0034
      IF (TK.LT.0.10) GO TO 13                                          PLTL0035
C  PRINT PLOT LABEL                                                     PLTL0036
```

```
      PRINT 101,       ETG(K),NB,EMU,AMBDA,ND,TANMU,CTO,SM,NG,IABC                PLTL0037
101   FORMAT ('1ETA=',F5.3,'      AIRLOADS(LB/IN) VS PSI(DEGREES)',               PLTL0038
     1 ' - THEORY = *, EXPERIMENT = X'/                                           PLTL0039
     2 1H ,I1,' BLADES  MU=',F5.3,'  LAMBDA=',F6.5,'  ND=',I1,                    PLTL0040
     3 '  MU*TAN(I)=',F6.5,'  CT=',F6.5,'  SIGMA=',F5.4,                          PLTL0041
     4 '    ITERATION=',I2,'    CASE=', I2/)                                      PLTL0042
C  PRINT LABEL FOR AIRLOADS AXIS                                                  PLTL0043
      PRINT 102,       (ELAB(I),I=1,13)                                           PLTL0044
102   FORMAT (' LDS=  ',F5.1,12(5X,F5.1))                                         PLTL0045
      GO TO 14                                                                    PLTL0046
13    PRINT 104,ETG(K)                                                            PLTL0047
104   FORMAT ('0ETA=',F5.3)                                                       PLTL0048
C  PRINT AIRLOADS AXIS                                                            PLTL0049
14    PRINT 103                                                                   PLTL0050
103   FORMAT (' PSI I',25('    I'))                                               PLTL0051
C  FILL ELPLT WITH BLANKS                                                         PLTL0052
      DO 15 I=1,128                                                               PLTL0053
15    ELPLT(I)=BLNK                                                               PLTL0054
C  START PSI LOOP                                                                 PLTL0055
      DO 20 J=1,NPSI                                                              PLTL0056
      IF (IPLX.EQ.0) GO TO 17                                                     PLTL0057
C  FIND LOCATION OF EXPERIMENTAL AIRLOADS POINT                                   PLTL0058
      I=TDEL*(XLDS(K,J)-ELMN(K))+6.5                                              PLTL0059
C  CHECK FOR POINTS BEYOND RANGE OF PLOT                                          PLTL0060
      IF (I.LT.1) I=1                                                             PLTL0061
      IF (I.GT.128) I=128                                                         PLTL0062
C  PUT X IN ELPLT FOR EXPERIMENTAL AIRLOADS POINT                                 PLTL0063
      ELPLT(I)=XPNT                                                               PLTL0064
      IX=I                                                                        PLTL0065
C  FIND LOCATION OF THEORETICAL AIRLOADS POINT                                    PLTL0066
17    I=TDEL*(AIRL(K,J)-ELMN(K))+6.5                                              PLTL0067
C  CHECK FOR POINTS BEYOND RANGE OF PLOT                                          PLTL0068
      IF (I.LT.1) I=1                                                             PLTL0069
      IF (I.GT.128) I=128                                                         PLTL0070
C  PUT * IN ELPLT FOR THEORETICAL AIRLOADS POINT                                  PLTL0071
      ELPLT(I)=STAR                                                               PLTL0072
```

```
C  PRINT PLOT LINE                                                          PLTL0073
      PRINT 105,         U(J),(ELPLT(N),N=1,128)                            PLTL0074
105   FORMAT (1H ,F4.0,128A1)                                               PLTL0075
C  RESET ELPLT TO BLANKS                                                    PLTL0076
      ELPLT(I)=BLNK                                                         PLTL0077
      ELPLT(IX)=BLNK                                                        PLTL0078
C  END PSI LOOP                                                             PLTL0079
20    CONTINUE                                                              PLTL0080
C  PRINT AIRLOADS AXIS                                                      PLTL0081
      PRINT 103                                                             PLTL0082
C  PRINT LABEL FOR AIRLOADS AXIS                                            PLTL0083
      PRINT 102,        (ELAB(I),I=1,13)                                    PLTL0084
C  END ETA LOOP                                                             PLTL0085
30    CONTINUE                                                              PLTL0086
      RETURN                                                                PLTL0087
      END                                                                   PLTL0088
```

```
      SUBROUTINE FXSQRT (M,N,DATA)                                      FXSQ0001
C  LDS-73                                                               FXSQ0002
      REAL I1,I2,LN                                                     FXSQ0003
      COMMON/SQVLS/I,J,K,KP,KPHI,L,IZ,                                  FXSQ0004
     1 XA,YA,ZA,XB,YB,ZB,ZM,VSM,DELL,GA,QT,QK,RHVS,DAM,QMO               FXSQ0005
      COMMON/BURST/IVLAC,IVCM,                                          FXSQ0006
     1 DVB,FVB,ETAC,DLXY,ZC,RHOC,RHOG,ABOER,ETA,AO,HRS,GK                FXSQ0007
      COMMON/SINCO/S(97),C(97),UTS(6,25)                                FXSQ0008
      COMMON/SGAM/NE,  NPSI, NPS,NG,   WF,EMU,A,BOER,DANG,              FXSQ0009
     1 SUM(6,25,18),ETG(6),DETA(6),ANG(25),PSI(25),ALPHA(6,25)           FXSQ0010
      COMMON/SBVGS/IG,MEP,GZER,EL(3),BV(25),GS(25)                      FXSQ0011
      COMMON/SQLDS/IABC,IEPS,IS,KZAD,NB,NL,ND,NDEL,NPOP,NPRP,NRNG,NDLM, FXSQ0012
     1 FTD,RHOK,DPHR,TANMU,AMBDA,                                        FXSQ0013
     2 ZETA(6),KPB(25),ELT(25),ELI(25),FTO(25)                           FXSQ0014
      COMMON/SAMB/NH,AMB(25,6),  AMBO(6),CLAM(9,6),SLAM(9,6)            FXSQ0015
      COMMON/SLDS/IRB,IRL,IPL,IPRF,IUT,NI,                              FXSQ0016
     1 CTO,SM,TH1,VT,EPS,LN,F,CON,ALPS,COSI,GMA,ELO,OMEPS,               FXSQ0017
     2 EMT,EMS,DENTH,A1MU,AZERO,AONE,BONE,ATWO,BTWO,THETAO,              FXSQ0018
     3 ETE(6),TC(6),ELDS(6),TW(6),GMC(6,25)                              FXSQ0019
      COMMON/SFX/PZAP,GPS,QTAN(6,25,25),                                FXSQ0020
     1 VI(3),VJ(3),VK(3),OVI(3),OVJ(3),OVK(3),D(3,25,97),QMAT(6,25,25)   FXSQ0021
      COMMON/SVS/  DX,DY,DZ,DXYS,CLS,CL,CA,CB,ZI,RAD,ZID,XI,YI,XK,YK,   FXSQ0022
     1 ZK,CAN,XH,YH,ZH,X1,Z,ZZ,X2,X2S,X1S,CAS,CBS,SOA,SOB,STA,STB,       FXSQ0023
     2 I1,I2                                                             FXSQ0024
      NAMELIST/DVS/DX,DY,DZ,DXYS,CLS,CL,CA,CB,ZI,RAD,ZID,XI,YI,XK,YK,   FXSQ0025
     1 ZK,CAN,XH,YH,ZH,X1,Z,ZZ,X2,X2S,X1S,CAS,CBS,SOA,SOB,STA,STB,       FXSQ0026
     2 I1,I2,GA,ABOER,IZ                                                 FXSQ0027
      NAMELIST/DUMP/I,J,K,KP,KPHI,L,PZAP,GPS,ETA,DELL,VSM,QK,QT,        FXSQ0028
     1 IVLAC,DVB,FVB,ETAC,DLXY,ZC,HRS,GK,RHVS,RHOC,RHOG,RHOK,DAM,        FXSQ0029
     2 XA,YA,ZA,XB,YB,ZB,ZM,VI,VJ,VK,OVI,OVJ,OVK,                        FXSQ0030
     3 IZ, IG,IS,IABC,IEPS,KZAD,MEP,NB,NE,NL,ND,NPSI,NDEL,NPOP,          FXSQ0031
     4 EMU,AMBDA,TANMU,AO,A,BOER,CTO,SM,VT,TH1,GZER,WF,DANG,DPHR,        FXSQ0032
     5 F,LN,CON,ALPS,ELO,A1MU,EPS,GMA,COSI,DENTH,OMEPS,EMT,EMS,          FXSQ0033
     6 AZERO,AONE,BONE,ATWO,BTWO,THETAO,                                 FXSQ0034
     7 S,C,ZETA,ETG,ETE,DETA,EL,ANG,PSI,TC,ELDS,                         FXSQ0035
     8 ALPHA,AMB,AMBO,CLAN,SLAM,GMC,UTS                                  FXSQ0036
```

```
      NAMELIST/PSUM/SUM,QMAT                                            FXSQ0037
      WRITE (6,DVS)                                                     FXSQ0038
      WRITE (6,DUMP)                                                    FXSQ0039
      WRITE (6,PSUM)                                                    FXSQ0040
      IF (ND.EQ.0) STOP                                                 FXSQ0041
      DO 10 JA=1,25                                                     FXSQ0042
      PRINT 100,JA                                                      FXSQ0043
  100 FORMAT ('1  J=',I2)                                               FXSQ0044
      DO 10 KA=1,NDLM                                                   FXSQ0045
      PRINT 101,     KA,(D(IA,JA,KA),IA=1,3)                            FXSQ0046
  101 FORMAT (' K=',I2,'  DIJK=',3E13.6)                                FXSQ0047
   10 CONTINUE                                                          FXSQ0048
      ITEST=2                                                           FXSQ0049
      IF (ITEST.EQ.2) STOP                                              FXSQ0050
      RETURN                                                            FXSQ0051
      END                                                               FXSQ0052
```

APPENDIX G

BLADE BENDING MODE SHAPE

The blade bending mode shape is chosen to be the simplest function which will satisfy the most important boundary conditions. The minimum power series polynomial for this purpose is a quadratic. For comparison, a quartic is also considered. The mode shape is a function of the radial mass distribution m($\bar{\eta}$) of the blade. Normally, this is assumed to be uniform outboard of the flapping hinge. The case of a concentrated tip mass on an otherwise uniform blade will also be considered.

For the quadratic case, it is assumed that m($\bar{\eta}$) is a constant, m (slug/ft). The assumed mode shape is:

$$\gamma_2(\bar{\eta}) = a\bar{\eta}^2 + b\bar{\eta} + c \tag{G.1}$$

The boundary conditions are:

1) Zero displacement at the flapping hinge:

$$\gamma_2(0) = 0.0 \tag{G.2}$$

2) Unit displacement at the blade tip:

$$\gamma_2(1.0) = 1.0 \tag{G.3}$$

3) Orthogonality with the rigid body mode:

$$\int_0^1 \eta\, \gamma_2(\bar{\eta})\, d\bar{\eta} = 0.0 \tag{G.4}$$

These conditions lead to:

$$\gamma_2(\bar{\eta}) = 4\bar{\eta}^2 - 3\bar{\eta} \quad \text{(G.5)}$$

For the quartic case $m(\bar{\eta})$ is again assumed to be a constant m (slug/ft). The assumed mode shape is:

$$\gamma_2(\bar{\eta}) = a\bar{\eta}^4 + b\bar{\eta}^3 + c\bar{\eta}^2 + d\bar{\eta} + e \quad \text{(G.6)}$$

Two extra boundary conditions can be included compared to the quadratic case:

4) Zero bending moment at the flapping hinge:

$$\left(\frac{d^2\gamma_2}{d\bar{\eta}^2}\right)_{\bar{\eta}=0} = 0.0 \quad \text{(G.7)}$$

5) Zero bending moment at the blade tip:

$$\left(\frac{d^2\gamma_2}{d\bar{\eta}^2}\right)_{\bar{\eta}=1.0} = 0.0 \quad \text{(G.8)}$$

The result is:

$$\gamma_2(\bar{\eta}) = \frac{-\bar{\eta}}{3}\left(10\bar{\eta}^3 - 20\bar{\eta}^2 + 7\right) \quad \text{(G.9)}$$

The $\gamma_2(\bar{\eta})$ shapes obtained from Eq. G.5 and Eq. G.9 are compared in Fig. G.1. There is very little difference, so the simpler quadratic expression (Eq. G.5) is normally used.

Finally, consider the case of a concentrated tip mass of M(slugs) on an otherwise uniform blade. The length and running mass of the blade outboard of the flapping hinge are L(ft) and m (slug/ft), respectively. Using a quartic $\gamma_2(\bar{\eta})$ and the same five boundary conditions:

$$Y_2(\bar{\eta}) = 10\bar{\eta}\left[\left(\frac{M}{mL}\right) - \frac{1}{3}\right]\left[\bar{\eta}^3 - 2\bar{\eta}^2 + \frac{30\left(\frac{M}{mL}\right) - 7}{30\left(\frac{M}{mL}\right) - 10}\right] \tag{G.10}$$

APPENDIX H

WAKE GEOMETRY PROGRAM WG-71

A listing of WG-71 is given in Appendix I. This appendix supplements that listing. The order of the input data is explained. Most of the data is input by NAMELIST. A set of lists defining the inputs in each NAMELIST is therefore given. If an input variable also appears in a COMMON statement this is noted. An annotated set of sample input data for WG-71 is given. The output is not included, due to its great bulk. Finally, there is an alphabetical list which defines the most important variables.

H.1 WG-71 Input Data

1) Read WDAT (real) at start of each case.

2) Read WCON (integer) at start of each case.

3) If (IG=1) Read ADETA

   Read $\alpha(\eta,\psi)$ from LDS-73.

```
      DO 10  K=1,NE
   10 READ 20, (ALPHA(K,J), J=1,NPSI)
   20 FORMAT (5E14.7)
```

4) If (I1 > 0) Read distortion $\vec{D}(\psi,\delta)$ as punched out by a previous WG-71 run.

```
      DO 10  K=1, NPPMI
   10 READ 20, ((D(I,J,K),I=1,3),J=1,NPSI)
   20 FORMAT (6E13.6)
```

5) If (CALCOMP Plot) Read DPLT

ILAB	If ILAB=2 no label on plot
SR	TPP scale factor, rotor radius is SR inches in plot
SK	Vertical scale factor, can be different from SR, if expanded vertical scale is desired

6) If (CALCOMP Plot and ILAB$\neq$0)

   then: Read plot labels (2 cards)

   Labels are the image of columns 1-60

7) If (CALCOMP Plot) Read ψ_s and number of ψ_s

```
      READ 10, J1, NS
   10 FORMAT (I3,I2)
```

$J1 = 1+(\psi_s/\Delta\psi)$ where ψ_s = starting azimuth of plotted wake = position of the η blade which shed plotted wake

NS = number of different ψ_s cases plotted; last (J1,NS) card has NS=1

Program expects a set of plot input cards [5, 6, 7 above] for every iteration of $\hat{D}$ plotted.

H.2 Namelists

H.2.1 NAMELIST/WDAT/

EMU	μ(COMMON/SSPLOT) - Advance ratio (normalized by tip speed)
AMBDA	λ(COMMON/SSPLOT) - Uniform, average downwash perpendicular to TPP - used to determine rigid wake geometry (normalized by tip speed)
TANMU	μtani(COMMON/SQCAL) - Inflow (downwash) perpendicular to TPP due to flight speed (relative wind) (normalized by tip speed)
ETA	η(COMMON/SSPLOT) - Radial station of point P_η, at which the distortion is computed normally $\eta = 1.0$ for tip vortex (normalized by R)
EL(6)	ℓ_ℓ (COMMON/SGAM) - Radial stations on ℓ blade which shed elements of the vortex wake. Up to 6 ℓ_ℓ can be used but normally only 3 are used, in which case they are defined as in LDS-73 with one exception. See definition of IABC in NAMELIST/WCOU (normalized by R)
RHOK	ρ_k(COMMON/SQCAL) - Nonburst value of ρ_c - vortex core radius (normalized by r)
RHOG	ρ_g(COMMON/SQCAL) - Burst value of ρ_c (normalized by R)
AO	a_o(COMMON/SQCAL) - Coning angle (zeroth harmonic of rigid body flapping) (radians)
SM	σ - Rotor solidity, $\sigma = \dfrac{\text{rotor blade area}}{\text{rotor disk area}} = \dfrac{n_b C}{\pi R}$
DPS1	$\Delta\psi$(COMMON/SGAM) - Azimuth angle interval between positions of the η blade, which sheds points P_η at which distortion is computed (degrees)

DM	Determines the assignment of wake elements to either the near wake or the far wake. If the magnitude of the contribution to the induced velocity at point P_η of a group of wake elements is $\lvert\vec{q}_s\rvert$, then, when $\lvert\vec{q}_s\rvert$ is greater than DM, those wake elements become part of the near wake relative to point P_η (normalized by tip speed) (Subsection 4.2.1)
QMO	(COMMON) - Minimum value of the induced velocity contribution of any vortex line or vortex sheet segment for which a special printout is made in QSVL, QCVL, or QVS (normalized by tip speed)
QMB	(COMMON) - If the induced velocity contribution of a bound vortex line is greater than QMB, integrate the induced velocity contribution over time (azimuth angle) ,as the blade sweeps through $\Delta\psi$ in closed form instead of assuming that the induced velocity contribution remains constant during $\Delta\psi$ (normalized by tip speed) (Subsection 4.1.8)
FWD	w_f (COMMON/SDCAL) - The increment in distortion ($\vec{D}$) during an azimuth interval $\Delta\psi$ is computed as FWD times the latest estimate plus (1.0-FWD) times the previous estimate (Subsection 4.2.4)
RHOI	ρ_I (COMMON/SWQC) - Vortex core radius of inboard trailing vortex line (normalized by R)
RHOL	ρ_L (COMMON/SWQC) - Vortex core radius of shed vortex lines (normalized by R)

H.2.2 NAMELIST/WCOU/

NI	Number of wake geometry iterations
NC	Number of cases - When a case is finished NC=NC-1 and if NC is still greater than zero, WDAT and WCOU are read in and a new case is started
IG	(COMMON/SGAM) - Indicates type of circulation computation used. IG = 0: Compute bound circulation $[\gamma_{bv}(\psi)]$ using angle-of-attack distribution $[\alpha(\eta,\psi)]$ left over from a previous case. IG = 1: Compute $\gamma_{bv}(\psi)$ using $\alpha(\eta,\psi)$ read in from cards along with $a = \partial C_L/\partial\alpha$, η, and $\Delta\eta$ in NAMELIST/ADETA

IG > 1: Set all $\gamma_{bv}(\psi) = \gamma_o = \dfrac{2C_T/\sigma}{\pi(1 - \ell_2^2)}$, no shed wake

IS (COMMON/SGAM) - Shed wake indicator

IS = 2: No shed wake

IS $\neq$ 2: Shed wake included

I1 (COMMON/SGAM) - Indicates initial wake model

I1 = 0: First iteration starting with rigid wake

I1 = n: nth iteration starting with $\vec{D}$ read in from a previous run of WG-71

IABC (COMMON/SGAM) - Indicates wake model used

VS = vortex sheet, VL = one vortex line, nVL = n vortex lines

IABC	Tip Vortex	Inboard Trailing Wake	Shed Wake
0	VL	VS	VL
1	VL	none	VL
2	VL	VL	VL
3	VL	nVL	VL
4	VL	VL	VS
5	VL	VS	VS

When IABC = 0 or 5, an inboard trailing vortex sheet model is used. The outboard edge of this vortex sheet is located at radial station ℓ_b. The value of ℓ_b is constant for all azimuth angles and is equal to either ℓ_2 or the average location of the radial peak in bound circulation, whichever is smaller (Subsection 4.1.2).

IPS Controls punch out of distortion $\vec{D}(\psi,\delta)$

IPS = 1: Punch only $\vec{D}$ for last iteration

IPS = 2: No punch

IPS = 3: Punch $\vec{D}$ for all iterations

IPR (COMMON/SDCAL) - Controls print out of intermediate $\vec{D}$ results before each general update.

IPR = 0: No intermediate print

IPR = n: Intermediate print for azimuth angles:

$\psi = 0, n\Delta\psi, 2n\Delta\psi, \ldots, 2\pi - n\Delta\psi$

IPLT	Controls CALCOMP plots of wake geometry IPLT = 0: No plots IPLT = 1: Plot only last iteration IPLT = 2: Plot all iterations
INPS	Controls print out of $\vec{D}$ at the end of each iteration except the last one. The $\vec{D}$ for the last iteration is always printed in full. If INPS=n: print $\vec{D}$ for azimuth angles: $\psi = 0, n\Delta\psi, 2n\Delta\psi, \ldots, 2\pi - n\Delta\psi$
NB	n_b (COMMON/SGAM) - Number of rotor blades
NL	n_ℓ (COMMON/SGAM) - Number of radial stations ℓ_ℓ on the ℓ blade. Use $n_\ell = 3$, except for IABC = 3 when n_ℓ = number of trailing vortex lines.
ND	(COMMON) - Value of subscript ℓ for which wake distortion ($\vec{D}$) is being computed. ND=NL for tip vortex
M	m - Number of turns of wake whose distortion ($\vec{D}$) is computed. This wake covers $2\pi m$ in age.
MB	M - Number of turns of wake below point P_η included in wake model for computing $\vec{D}$ at point P_η. If the age of point P_η is δ then the total azimuthal extent of the wake model is $\delta + 2\pi M$ (Subsection 4.1.3).
LDM	ℓ_{DM} (COMMON/SDCAL) - General updating parameter. Every ℓ_{DM} steps $\Delta\psi$ in the boundary age δ_M a general (distortion) update is executed (Subsection 4.2.3)
NDM(25)	$n_{DM}(\psi)$ (COMMON/SGAM) - Boundary updating parameter. At a given azimuth angle ψ a boundary (induced velocity) update is executed every $n_{DM}(\psi)$ steps $\Delta\psi$ in the age δ (Subsection 4.2.2).
KPB(25)	(COMMON/SQCAL) - Tip vortex core bursting occurs at azimuth ϕ when point P_ℓ reaches age $\delta \geq \delta_b(\phi)$. $KPB(\phi) = 1 + \delta_b(\phi)/\Delta\psi$ [output by LDS-73]

H.2.3 NAMELIST/ADETA/

A	$a = \frac{\partial C_L}{\partial \alpha}$ [Same as LDS-73]

NE n_η - Number of radial stations in angle of attack distribution $\alpha(\eta,\psi)$ input for circulation computation [same as LDS-73]

ETG(10) η_k - Radial stations of $\alpha(\eta,\psi)$ [same as LDS-73]

DETA(10) $\Delta\eta_k$ - Radial integration intervals for $\alpha(\eta,\psi)$ [same as LDS-73]

H.3 Sample Input Data for WG-71

& WDAT

EMU = .18

AMBDA = .0243 [same as LDS-73]

TANMU = .009

ETA = 1.0 [tip vortex]

EL = .10, .85, 1.0 EL(2) = .85 is the max. allowable location of the outboard edge of the inboard trailing vortex sheet since IABC = 0 for this case

AO = .085

SM = .0622 [same as LDS-73]

DPS1 = 15.

DM = .0020 [use (.04 → .08)(λ - μtani)]

RHOK = .0025

RHOG = .10

[same as LDS-73 with IVCM = 1]

RHOL = .105 [use ∿ 0.4 $\Delta\psi$]

RHOI = .30 [use ∿ 0.5 ($\ell_2 - \ell_1$)]

QMO = .015 [use (.5 → 1.0)(λ - μtani)]

QMB = .0020 [about the same as DM]

FWD = .5

& WCOU

NI = 3 [sometimes 2 iterations is enough]

I1 = 0 [start with rigid wake]

NC = 1

IG = 1 [input $\alpha(\eta,\psi)$]

IS = 1 [include shed wake - it makes a difference]

IABC = 0 [normally best choice]

IPS = 1 [punch $\vec{D}$ for last iteration only]

IPR = 12 [intermediate print for ψ = 0°, 180°]

IPLT = 2 [plot every iteration to check convergence]

INPS = 6 [print $\vec{D}$ at ψ = 0°, 90°, 180°, 270° for all but last iteration]

NB = 4 [4 bladed rotor]

NL = 3 [IABC = 0]

ND = 3 [tip vortex (ND = NL)]

M = 2

MB = 2 [$m \sim 0.4/\mu$ is about right]

LDM = 12 general updating is expensive - not done very frequently - every 180° works well normally

NDM = 6,6,6,7*3,6*6,7*3,6,6,

boundary updating is less expensive - done every 90° fore and aft and every 45° along the sides of rotor because there is more tip vortex interaction along the sides

KPB = 7*49,4*6,5,9,9,10,6*49 [printed out by LDS-73]

<u>& ADETA</u>

A 5.73

NE = 6

ETG = .25,.40,.55,.75,.85,.95, [same as LDS-73]

DETA = .225,.15,.225,.10,.10,.10

α cards from LDS-73 with η cards removed

<u>& DPLT</u>

ILAB = 1 [read in plot labels]

SR = 2 [scale factor in TPP → radius = 2 inches]

SK = 2 [vertical scale factor]

The next two cards are the image of the two line label at the top of each CALCOMP plot. Anything in columns 1-60 of these two cards will appear on the plots.

The next cards are read by READ 10,J1,NS

```
                                  10 FORMAT (I3,I2)
```

J1 is the subscript of the azimuthal position of the blade for which a CALCOMP plot is made. For example: $J1=25 \rightarrow \psi = 360°$, $J1 = 13 \rightarrow \psi = 180°$ NS controls the number of different starting ψ's read in. When $NS > 1$, the program will try to read in another starting ψ card after it finishes the current plot.

There is a complete set of plot cards, starting with the scale factor card and ending with the starting ψ (blade position) cards, for each iteration whose $\vec{D}$ is plotted.

H.4 WG-71 Outputs

H.4.1 Printed Output

The printed output is presented in the order in which it is output.

1) Label plus input data from WDAT and WCOU - output by MAIN
2) Bound circulation $\gamma_{bv}(\psi)$ and ℓ_2 - output by WGAM
3) Iteration number - output by MAIN
4) Special print for any vortex line or vortex sheet segment whose contribution to the induced velocity at point P_η is greater than QMO - output by QSVL, QCVL, and QVS
5) Intermediate distortion results, output before each general update. At this point in the computation $KDLL = 1 + \delta_M/\Delta\psi$. The index N refers to the vector components: $N = 1 - \vec{i}$, $N = 2 - \vec{j}$, $N = 3 - \vec{k}$. The index J gives the azimuth angle: $J = 1 + \psi\ \Delta\psi$. These results are only output for selected ψ controlled by the input IPR. The unlabeled numbers are $\vec{D}(\psi,\delta)$ starting at $\delta = \Delta\psi$ and going to $\delta = 2\pi m$. The labeled numbers are: $DS = \vec{D}_s(\psi)$, $DSN = \vec{D}_N(\psi)$, $DVO = \vec{D}_o(\psi)$, $CUR = \vec{q}_D(\psi)$, $VEL = \vec{q}_F(\psi,\ \delta_M)$, and $QBV = \vec{q}_{BV}$ - output by DCALC.

6) Transition points KTR = $\delta_T(\zeta,\psi, k)/\Delta\psi$. Under each PSI = ψ the KTR for the different values of ζ are listed in rows: 1st row - $\zeta = 0$, 2nd row - $\zeta = \Delta\zeta$, etc. Within a row, the numbers are in order by k, from k = 1 to $k_m(\zeta,\psi)$ - output by MAIN.

7) Main output of $\vec{D}(\psi,\delta)$ and $\vec{q}_F(\psi,\delta)$. Each page is for a given value of azimuth angle PSI = ψ or starting azimuth angle PSI - S = $\psi_s = \psi + \delta$. Column 1 is the age DELTA = δ. The next 6 columns are for the given ψ. Columns 2-4 are the $\vec{i}$, $\vec{j}$, and $\vec{k}$ components of D(PSI) = $\vec{D}(\psi,\delta)$. Columns 5-7 are the $\vec{i}$, $\vec{j}$, and $\vec{k}$ components of VEL(PSI) = $\vec{q}_F(\psi,\delta)$. The remaining 7 columns are for the given $\psi_s = \psi + \delta$. These columns are useful for plotting the distortion of a given tip vortex line, because they are all for the same ψ_s. The starting azimuth angle ψ_s is the position of the η blade at age δ. Hence it is where the tip vortex line starts. Column 8 is $\psi = \psi_s - \delta$, where δ comes from column 1 and ψ_s from the top of the page. Columns 9-11 are the $\vec{i}, \vec{j}$, and $\vec{k}$ components of $\vec{D}(\psi,\delta)$, where ψ comes from column 8. Column 12 (DK) is column 11 plus $\delta\mu$tani. This is the actual vertical displacement of point P_η since age zero. Column 13 (VK) is the average downwash required to give column 12(DK), VK = DK/δ. Column 14 (WK) is the average downwash required to give column 4 WK = $D_k(\psi,\delta)/\delta$ - output by MAIN.

8) Execution times in hundredths of a second - output by MAIN.

9) Repeat numbers 3 through 8 for each iteration.

H.4.2 Punched Output

The only punched output is the distortion $\vec{D}(\psi,\delta)$. This data can be used as input to LDS-73 for a distorted wake run or to WG-71 for more distortion iterations. Normally, $\vec{D}(\psi,\delta)$ is only punched out for the last iteration; however, this can be controlled by input IPS.

```
          DO    10    K=1, NPHI
   10     PUNCH 20,((D(I,J,K),I=1,3),J=1NPSI)
   20     FORMAT (6E13.6)
```

H.4.3 CALCOMP Plots

Subroutine SPLOT makes Calcomp plots of $D(\psi,\delta)$. The input IPLT controls the calling of SPLOT. Subroutine SPLOT reads in two lines of labels, up to 60 characters long, which are printed at the top of each plot. It also reads values for the starting azimuth angle ψ_s. There is a separate plot for each ψ_s. Each plot consists of a top view (perpendicular to the TPP) and a side view of the tip vortex of the η blade from $\delta = 0$ to $\delta = 2\pi m$. Both the rigid wake and the distorted wake versions of the tip vortex are shown. For an example, see Fig. 1.

H.5 Selected Variables Used by WG-71

To aid the reading of the listing of WG-71 in Appendix I, the most important variables are defined here. When possible, a variable is defined by reference to symbols defined in the List of Symbols or by reference to the input namelists WDAT, WCOU, and ADETA (Subsection H.2).

A	$a = \frac{\partial C_L}{\partial \alpha}$ (input ADETA)
ALPHA(10,25)	$\alpha(\eta,\psi)$ - input from LDS-73
AMBDA	λ (input WDAT)
ANG(25)	ψ (degrees)
AO	a_o (input WDAT)
AXBI,AXBJ,AXBK	$(axb)_i$, $(axb)_j$, $(axb)_k$
AXBS	$\lvert\vec{a}x\vec{b}\rvert$
BOER	b/2R
BVG(6,25)	$\frac{b}{2R}\gamma_{bv}(\ell,\phi)$
CUR(3,25)	$\vec{q}_o(\psi)$
D(3,25,97)	$\vec{D}(\psi,\delta)$
DANG	$\Delta\psi$ (radians)
DELL	Δx
DELTA	δ
DETA(10)	$\Delta\eta_k$ (input ADETA)
DM	(input WDAT)
DPSI	$\Delta\psi$ (degrees) (input WDAT)

DS(3,25)	$\vec{D}_s(\psi)$
DSMI,DSMJ,DSMK	i,j, and k components of Δq
DSN(3,25)	$\vec{D}_N(\psi)$
DVO(3,25)	$\vec{D}_o(\psi)$
D3(3)	$\vec{D}_\eta(\psi,\delta)$
EL(6)	ℓ_ℓ (input WDAT)
EMU	μ (input WDAT)
ETA	η (input WDAT)
ETACP	$\eta \cos \psi$
ETASP	$\eta \sin \psi$
ETG(10)	η_k (input ADETA)
FWD	w_f (input WDAT)
GA, GB	γ_a, γ_b
GBV(6,25)	$\frac{b}{2R} \gamma_t(\ell,\phi)$
GS(6.25)	$\frac{b}{2R} \gamma_s(\ell,\phi)$
GZER	$\gamma_o = \frac{2(C_T/\sigma)}{\pi(1 - \ell_2^2)}$
I	$1 + \zeta/\Delta\zeta$
IABC	(input WCOU)
IG	(input WCOU)
INPS	(input WCOU)
IPLT	(input WCOU)
IPR	(input WCOU)
IPS	(input WCOU)
IS	(input WCOU)
I1	(input WCOU)
J	$1 + \psi/\Delta\psi$
K	k in $\delta_T(\psi, \zeta, k)$
KDLL	$\delta_M/\Delta\psi$ normally, however KDLL = $1 + \delta_M/\Delta\psi$ just before a general update
KDLS	$1 + \delta_M/\Delta\psi$

KM(6,25)	$k_m(\zeta,\psi)$
KP	$1 + \delta_\ell/\Delta\psi$
KPB(25)	$1 + \delta_b(\phi)/\Delta\psi$ (input WCOU)
KPHI	$1 + \phi/\Delta\psi$, principal value only $1 \leq$ KPHI $\leq$ NPSI
KSTRT	$1 + \delta_{ST}/\Delta\psi$
KT	Value of KTR for end of current near wake section of wake
KTR(6,25,16)	$\delta_T(\zeta, \psi, k)/\Delta\psi$
L	Index ℓ from ℓ_ℓ
LD	$1 + \delta_D/\Delta\psi$
LDM	ℓ_{DM} (input WCOU)
LG	L - 1
LL	$1 + \delta/\Delta\psi$
LL1	$1 + \delta_1/\Delta\psi$
LS	$1 + \delta_s/\Delta\psi$
M	m (input WCOU)
MB	M (input WCOU)
MGO	Indicates current state of distortion iteration, See Subsection 4.2.5.
NB	n_b (input WCOU)
NBIGM	$2\pi M/\Delta\psi$
NC	(input WCOU)
ND	(input WCOU)
NDI	$\Delta\zeta/\Delta\psi$
NDM(25)	$n_{DM}(\psi)$ (input WCOU)
NE	n_η (input ADETA)
NGO	Indicates curved tip vortex-line segment when NGO = 2
NI	(input WCOU)
NL	n_ℓ (input WCOU)

NPHI	$1 + 2\pi m/\Delta\psi$ - number of age stations δ
NPSI	$n_\psi = 1 + 2\pi/\Delta\psi$
OKPHI	$1 + (\phi - \Delta\psi)/\Delta\psi$, principal value only $1 \leq$ OKPHI $\leq$ NPSI
OLD(3)	$\Delta\vec{D}_o$
OLDM	When OLDM = 0, there is a boundary update; i.e., $(\delta - \Delta\psi)/\Delta\psi$ is an integral multiple of $n_{DM}(\psi)$
OVI(6),OVJ(6), OVK(6)	Vectors a or b from point P_η to various points P_ℓ as a function of ℓ_ℓ for the previous age $\delta_\ell + \Delta\psi$
PHI	ϕ (radians)
PSI(25)	ψ (radians)
PZAP	$\psi + \zeta - \phi = \delta_\ell - \delta = \delta_R$
QBV(3)	$\vec{q}_{BV}$
QMB	(input WDAT)
QMO	(input WDAT)
PHOC	ρ_c
RHOG	ρ_g (input WDAT)
RHOI	ρ_I (input WDAT)
RHOK	ρ_k (input WDAT)
RHOL	ρ_L (input WDAT)
SKPHI	$1 + (\phi - 2\Delta\psi)/\Delta\psi$, principal value only $1 \leq$ SKPHI $\leq$ NPSI
SM	σ (input WDAT)
START	$2\pi M$
SUMI,SUMJ,SUMK	$\vec{i}$, $\vec{j}$, and $\vec{k}$ components of $\Sigma\vec{q}$
SVI(6),SVJ(6), SVK(6)	Same as OVI(6), etc., but for age $\delta_\ell + 2\Delta\psi$, used for curved-tip vortex-line segments.
TAMB	$\lambda - \mu$tani
TANMU	μtani (input WDAT)
TOT(3,25)	$\vec{q}_{TOT}(\psi)$
VEL(3,25,97)	$\vec{q}_F(\psi,\delta)$
VI(6),VJ(6), VK(6)	Vectors $\vec{a}$ or $\vec{b}$ from point P_η to various points P_ℓ as a function of ℓ_ℓ for the current δ_ℓ
XA,YA,ZA	x_a, y_a, z_a

XB,YB,ZB	x_b, y_b, z_b
ZETA	ζ
ZM	z_m

APPENDIX I

LISTING OF WG-71

This program is normally compiled and executed on an IBM 370/168 running under release 21.7 of OS/MVT (Operating System/Multiprogramming with a Variable number of Tasks) with release 21.6 of FORTRAN G1. The core storage requirement is 200k. Subroutine SPLOT make plots of the tip vortex geometry on a CALCOMP 563 Drum Plotter with a CALCOMP 905 Controller. If no plots are required remove statements MAIN0039, MAIN0040, MAIN0162, MAIN0163, MAIN0164, MAIN0165, MAIN0166, MAIN0189, and Subroutines SETCAL and SPLOT.

```
C  WG 71                                                                  MAIN0001
C   INCLUDES  PROVISION  FOR  VORTEX  LINES  AND  SHEETS                   MAIN0002
      INTEGER OKPHI,SKPHI                                                 MAIN0003
      COMMON XA,YA,ZA,XB,YB,ZB,ZM,DELL,GA,GB,RHOC,RHOS,QI,QJ,QK,          MAIN0004
     1 IEG,I,J,K,KP,KPHI,KT,KDLL,L,LD,LG,LL,MGO,NGO,ND,KPO,KSTRT,         MAIN0005
     2 QMO,QMB,SQ                                                         MAIN0006
      COMMON/SSPLOT/EMU,AMBDA,ETA,C(97),S(97),PSI(97),D(3,25,97)          MAIN0007
      COMMON/SGAM/IABC,I1,IG,IS,LST,NL,NB,NPSI,NPHI,NIN,NOUT,NDI,         MAIN0008
     1 EM,DPSI,DANG,TAMB,GZER,BOER,                                       MAIN0009
     2 EL(6),ELB(25),ANG(97),ZETA(6),GBV(6,25),BVG(6,25),GS(6,25),        MAIN0010
     3 ENDM(25),NDM(25)                                                   MAIN0011
      COMMON/SQCAL/OKPHI,SKPHI,OLDM,PHI,SUMI,SUMJ,SUMK,                   MAIN0012
     1 OSUMI,OSUMJ,OSUMK,TANMU,TAMD,PZA,PZAP,AO,ETASP,ETACP,DELTA,DMS,    MAIN0013
     2 RHOK,RHOG,START,NBIGM,                                             MAIN0014
     1 AD(3),D3(3),VI(6),VJ(6),VK(6),OVI(6),OVJ(6),OVK(6),                MAIN0015
     2 DVO(3,25),DS(3,25),DSN(3,25),TOT(3,25),CUR(3,25),QBV(3),           MAIN0016
     3 KM(6,25),KTR(6,25,16),VEL(3,25,97),KPB(25)                         MAIN0017
      COMMON/SDCAL/IPR,NPSR,LDM,FWD,EWD,DSMI,DSMJ,DSMK,DSMS               MAIN0018
      COMMON/SWOC/LIB,RHOI,RHOL,RHO                                       MAIN0019
      NAMELIST/WDAT/EMU,AMBDA,TANMU,ETA,EL,RHOK,RHOG,AO,SM,               MAIN0020
     1 DPSI,DM,QMO,QMB,FWD,RHOI,RHOL                                      MAIN0021
     2 /WCOU/NI,NC,IG,IS,I1,IABC,IPS,IPR,IPLT,INPS,                       MAIN0022
     3 NB,NL,ND,M,MB,LDM,NDM,KPB                                          MAIN0023
     4 /TIME/ITDAT,ITGAM,ITSTRT,ITEND,ITEX,ITPLT,                         MAIN0024
     5 IODAT,IOGAM,IOSTRT,IOEND,IOPLT                                     MAIN0025
      EXTERNAL FXOF                                                       MAIN0026
      CALL ERRSET (207,0,0,0,FXOF)                                        MAIN0027
      CALL ERRSET (208,0,0,0,FXOF)                                        MAIN0028
      CALL ERRSET (209,0,0,0,FXOF)                                        MAIN0029
      NIN=5                                                               MAIN0030
      NOUT=6                                                              MAIN0031
      MPLT=1                                                              MAIN0032
 1    CONTINUE                                                            MAIN0033
C  START NEW CASE                                                         MAIN0034
      CALL TIMING (ITDAT,IODAT)                                           MAIN0035
C  READ INPUT DATA                                                        MAIN0036
```

```
      READ (NIN,WDAT)                                                   MAIN0037
      READ (NIN,WCOU)                                                   MAIN0038
      NPLT=IPLT*MPLT                                                    MAIN0039
      IF (NPLT.GT.0) CALL SETCAL (MPLT)                                 MAIN0040
C  COMPUTE CONSTANTS                                                    MAIN0041
      NPSI=360.1/DPSI                                                   MAIN0042
      NDI=NPSI/NB                                                       MAIN0043
      NPHI=M*NPSI+1                                                     MAIN0044
      NBIGM=MB*NPSI                                                     MAIN0045
      NPSI=NPSI+1                                                       MAIN0046
      NPSR=NPSI-IPR                                                     MAIN0047
      TAMB=AMBDA-TANMU                                                  MAIN0048
      CT=2.0*TAMB*SQRT(EMU*EMU+AMBDA*AMBDA)                             MAIN0049
      CTSM=CT/SM                                                        MAIN0050
      ENB=NB                                                            MAIN0051
      BOER=0.7854*SM/ENB                                                MAIN0052
      GZER=CTSM/(1.5708*(1.0-EL(2)**2))                                 MAIN0053
      BIGM=MB                                                           MAIN0054
      EM=M                                                              MAIN0055
      START=6.2831853*BIGM                                              MAIN0056
C  WRITE INPUT DATA                                                     MAIN0057
      WRITE (NOUT,2) EMU,SM,DPSI,LDM,NDM(1),                            MAIN0058
     1 TANMU,BOER,MB,M,IABC,                                            MAIN0059
     2 AMBDA,RHOK,NPSI,NPHI,NI,I1,                                      MAIN0060
     3 ETA,RHOG,DM,IG,                                                  MAIN0061
     4 EL(2),CT,QMB,IS,                                                 MAIN0062
     5 AO,CTSM,OMO,IPS,IPR,                                             MAIN0063
     6 NB,GZER,NL,ND,FWD                                                MAIN0064
2     FORMAT ('1WAKE` GEOMETRY  71'//                                   MAIN0065
     1 'OMU=',F5.3,8X,'SIGMA=',F5.3,'   DPSI=',                         MAIN0066
     1 F4.1,5X,'L/NDM=',I2,'/',I1/                                      MAIN0067
     2 'OTANMU=',F7.4,'   B/2R=',F6.4,'   BM/M=',                       MAIN0068
     2 I1,'/',I1,6X,'IABC=',I1/                                         MAIN0069
     3 'OLAMBDA=',F6.3,'   RHOK=',F6.4,'   NPS/H=',                     MAIN0070
     3 I2,'/',I2,'   NI/I1=',I1,'/',I1/                                 MAIN0071
     4 'OETA=',F5.3,7X,'RHOG=',F6.4,'   DM=',                           MAIN0072
```

```
     4 F7.5,'     IG=',I1/
     5 'OL(2)=',F5.3,6X,'CT=',F7.5,'     QMB=',
     5 F6.4,'     IS=',I1/
     6 'OA-O=',F6.4,6X,'CT/SM=',F5.3,
     6 '    QMO=',F6.4,'     IPS/R=',I1,'/',I2/
     7 'ONB=',I1,12X,'GZER=',F6.4,'    NL/ND=',
     7 I1,'/',I1,5X,'FWD=',F4.2)
      WRITE (NOUT,3) (NDM(J),J=1,NPSI)
3     FORMAT ('ONDM(J)  J= 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
     116 17 18 19 20 21 22 23 24 25'/'        NDM=',I2,24I3)
      WRITE (NOUT,4) (KPB(J),J=1,NPSI)
4     FORMAT ('        KPB=',I2,24I3)
C  COMPUTE CONSTANTS
      DANG=.1745329E-1*DPSI
      TAMD=TAMB*DANG
      BODEP=BOER*DANG
      TAD=-TANMU*DANG
      ELDM=LDM
      DMS=DM*DM
      NG=I1
      EWD=1.0-FWD
      CALL TIMING (ITGAM,IOGAM)
C  COMPUTE CIRCULATION (GAMMA)
C  SET UP SIN, COS, AND ZETA ARRAYS
C  INITIALIZE DISTORTION (D)
      CALL WGAM
      LIB=LST
      IF (IABC.LE.0) LIB=1
C  START NEW ITERATION
14    CONTINUE
      CALL TIMING (ITSTRT,IOSTRT)
      NG=NG+1
      WRITE (NOUT,5) NG
5     FORMAT ('1START ITERATION NUMBER',I2)
C   INITIALIZE
      DO 15 N=1,3
```

```
      DO 15 J=1,NPSI
      DS(N,J)=0.0
      DSN(N,J)=0.0
      VEL(N,J,1)=0.0
15    VEL(N,J,2)=0.0
C   START LOOP SYSTEM
C  COMPUTATION OF WAKE DISTORTION D(N,J,LL)
      CALL DCALC
C   HOORAY---HAVE NOW COMPUTED A TABLE OF D
      CALL TIMING (ITEND,IOEND)
      ITEX=ITEND-ITSTRT
C  PRINT OUT TRANSISTION POINTS
      WRITE (NOUT,452)
452   FORMAT ('1TRANSITION  POINTS  KTR')
      DO 455 J=1,NPSI
      WRITE (NOUT,453) ANG(J)
453   FORMAT (' PSI=',F4.0)
      DO 455 I=1,NB
      KMIJ=KM(I,J)
      WRITE (NOUT,454) (KTR(I,J,K),K=1,KMIJ)
454   FORMAT (' KTR=',16I4)
455   CONTINUE
C  PRINT OUT DISTORTION
      IF (NI.EQ.1) INPS=1
      DO 511 J=1,NPSI,INPS
      WRITE (NOUT,512) ANG(J)
      JK=J
      K=1
      ADD=0.0
      DK=0.0
      VLK=0.0
      WK=0.0
510   CONTINUE
C  D(I,J,K) GIVES TIME HISTORY OF DISTORTION AT GIVEN PSI
C  D(I,JK,K) GIVES SNAPSHOT OF DISTORTION AT GIVEN PSI-START
      WRITE (NOUT,513) ANG(K),(D(I,J,K),I=1,3),(VEL(I,J,K),I=1,3),
```

```
     1 ANG(JK),(D(I,JK,K),I=1,3),DK,VLK,WK                             MAIN0145
      K=K+1                                                             MAIN0146
      IF (K.GT.NPHI) GO TO 511                                          MAIN0147
      JK=JK-1                                                           MAIN0148
      IF (JK.EQ.0) JK=NPSI-1                                            MAIN0149
      ADD=ADD-TAD                                                       MAIN0150
C  DK ADDS MU*TAN(I) TO D(3,JK,K)                                       MAIN0151
      DK=ADD+D(3,JK,K)                                                  MAIN0152
C  VLK IS AVERAGE LAMBDA TO GIVE DK                                     MAIN0153
      VLK=DK/PSI(K)                                                     MAIN0154
C  WK IS AVERAGE LAMBDA TO GIVE D(3,J,K) - NO MU*TAN(I)                 MAIN0155
      WK=D(3,J,K)/PSI(K)                                                MAIN0156
      GO TO 510                                                         MAIN0157
511   CONTINUE                                                          MAIN0158
512   FORMAT ('1PSI(PSI-S)=',F4.0/'ODELTA',11X,'D(PSI)',22X,            MAIN0159
     1 'VEL(PSI)',14X,'PSI',11X,'D(PSI-S)',14X,'DK',8X,'VK',8X,'WK'/)   MAIN0160
513   FORMAT (1X,F5.0,3F9.5,2X,3F9.5,4X,F5.0,3F9.5,3F10.5)              MAIN0161
      IF (IPLT.EQ.0) GO TO 549                                          MAIN0162
      IF (IPLT.EQ.1.AND.NI.GT.1) GO TO 549                              MAIN0163
C  CALL CALCOMP PLOT ROUTINE                                            MAIN0164
      CALL SPLOT (NPHI,NPSI,TAD,DANG)                                   MAIN0165
549   CONTINUE                                                          MAIN0166
C  PRINT OUT TIME                                                       MAIN0167
      CALL TIMING (ITPLT,IOPLT)                                         MAIN0168
      WRITE (NOUT,TIME)                                                 MAIN0169
C   NOW TEST FOR ANOTHER ITERATION                                      MAIN0170
      NI=NI-1                                                           MAIN0171
      IF(NI.LE.0)     GO TO 550                                         MAIN0172
      I1=I1+1                                                           MAIN0173
      IF (IPS.LE.2) GO TO 14                                            MAIN0174
C  PUNCH OUT DISTORTION                                                 MAIN0175
      DO 5495 K=1,NPHI                                                  MAIN0176
      WRITE (7,551) ((D(I,J,K),I=1,3),J=1,NPSI)                         MAIN0177
5495  CONTINUE                                                          MAIN0178
      GO TO 14                                                          MAIN0179
550   NC=NC-1                                                           MAIN0180
```

```
      IF (IPS.EQ.2) GO TO 555
      DO 552 K=1,NPHI
      WRITE (7,551) ((D(I,J,K),I=1,3),J=1,NPSI)
  551 FORMAT (6E13.6)
  552 CONTINUE
  555 CONTINUE
C   TEST FOR MORE CASES
      IF(NC.GT.0)   GO TO 1
      IF (IPLT.GT.0) CALL ENDPLT (0.0,0.0,999)
      STOP
      END
```

```
      SUBROUTINE SETCAL (MPLT)                                          STCL0001
      MPLT=0                                                            STCLC002
      CALL PLOTS (IDUM,IDUM,9)                                          STCL0003
      RETURN                                                            STCL0004
      END                                                               STCL0C05
```

```
      SUBROUTINE WGAM                                                   WGAM0001
C  WG 71                                                                WGAM0002
C  SET UP SIN, COS, AND ZETA ARRAYS                                     WGAM0003
C  COMPUTE CIRCULATION (GAMMA)                                          WGAM0004
C  INITIALIZE DISTORTION (D)                                            WGAM0005
      DIMENSION ALPHA(10,25),ETG(10),DETA(10),BVO(6,25)                 WGAM0006
      COMMON/SSPLOT/EMU,AMBDA,ETA,C(97),S(97),PSI(97),D(3,25,97)        WGAM0007
      COMMON/SGAM/IABC,I1,IG,IS,LST,NL,NB,NPSI,NPHI,NIN,NOUT,NDI,       WGAM0008
     1 EM,DPSI,DANG,TAMB,GZER,BOER,                                     WGAM0009
     2 EL(6),ELB(25),ANG(97),ZETA(6),GBV(6,25),BVG(6,25),GS(6,25),      WGAM0010
     3 ENDM(25),NDM(25)                                                 WGAM0011
      NAMELIST/ADETA/A,NE,ETG,DETA                                      WGAM0012
C   SET UP TABLES OF ZETA,PSI,ETC. AFTER DEFINING NECESSARY COUNTERS    WGAM0013
      ZETA(1)=0.                                                        WGAM0014
      ZADD=6.2831853/NB                                                 WGAM0015
      DO 7 I=2,NB                                                       WGAM0016
  7   ZETA(I)=ZETA(I-1)+ZADD                                            WGAM0017
      PSI(1)=0.0                                                        WGAM0018
      C(1)=1.0                                                          WGAM0019
      S(1)=0.0                                                          WGAM0020
      ANG(1)=0.                                                         WGAM0021
      DO 8 J=2,NPHI                                                     WGAM0022
      ANG(J)=ANG(J-1)+DPSI                                              WGAM0023
      PSI(J)=PSI(J-1)+DANG                                              WGAM0024
  8   CONTINUE                                                          WGAM0025
      KKK=NPSI+1                                                        WGAM0026
      NEM=EM-.999                                                       WGAM0027
      CD= COS(DANG)                                                     WGAM0028
      SD= SIN(DANG)                                                     WGAM0029
      ENDM(1)=NDM(1)                                                    WGAM0030
      DO 9 J=2,NPSI                                                     WGAM0031
      ENDM(J)=NDM(J)                                                    WGAM0032
      JG=J-1                                                            WGAM0033
      C(J)=CD*C(JG)-SD*S(JG)                                            WGAM0034
      S(J)=CD*S(JG)+SD*C(JG)                                            WGAM0035
      DO 9 M=1,NEM                                                      WGAM0036
```

```
      JM=J+M*(NPSI-1)                                                   WGAM0037
      C(JM)=C(J)                                                        WGAM0038
      S(JM)=S(J)                                                        WGAM0039
9     CONTINUE                                                          WGAM0040
      ELBM=EL(2)                                                        WGAM0041
C  CIRCULATION (GAMMA) CALCULATION                                      WGAM0042
C  BVG = BOUND CIRCULATION                                              WGAM0043
C  GBV = TRAILING WAKE CIRCULATION                                      WGAM0044
C  GS = SHED WAKE CIRCULATION                                           WGAM0045
      IF (IG.GT.1)  GO TO 11                                            WGAM0046
      IF (IG.EQ.0) GO TO 104                                            WGAM0047
C  READ IN ANGLE OF ATTACK DISTRIBUTION FROM AIRLOADS COMPUTATION       WGAM0048
      READ (NIN,ADETA)                                                  WGAM0049
      DO 102  I=1,NE                                                    WGAM0050
102   READ(NIN,103) (ALPHA(I,J),J=1,NPSI)                               WGAM0051
103   FORMAT (5E14.7)                                                   WGAM0052
104   CONTINUE                                                          WGAM0053
C  ALL GAMMA MULTIPLIED BY B/2R                                         WGAM0054
      AP=BOER*A/360.0                                                   WGAM0055
      IF(IABC.EQ.5)    GO TO 110                                        WGAM0056
      IF(IABC.EQ.3)   GO TO 114                                         WGAM0057
C  VORTEX LINE WAKE MODEL WITH 1 OR 2 TRAILING VORTEX LINES             WGAM0058
      DO 106 J=1,NPSI                                                   WGAM0059
      EMS=EMU*S(J)                                                      WGAM0060
      GAM=0.0                                                           WGAM0061
      DO 105 I=1,NE                                                     WGAM0062
      GA=(ETG(I)+EMS)*ALPHA(I,J)                                        WGAM0063
105   GAM=AMAX1(GAM,GA)                                                 WGAM0064
106   GBV(3,J)=AP*GAM                                                   WGAM0065
      IF(IS.EQ.2)   GO TO 108                                           WGAM0066
      NPS=NPSI-1                                                        WGAM0067
      DO 107 J=1,NPS                                                    WGAM0068
107   GS(3,J)= GBV(3,J+1)-GBV(3,J)                                      WGAM0069
      GS(3,NPSI)=GS(3,1)                                                WGAM0070
108   CONTINUE                                                          WGAM0071
      LST=2                                                             WGAM0072
```

```
1085  DO 109 J=1,NPSI                                                   WGAM0073
      GBV(2,J)=-GBV(3,J)                                                WGAM0074
      BVG(3,J)=GBV(3,J)                                                 WGAM0075
      BVO(3,J)=BVG(3,J)/BDER                                            WGAM0076
      ELB(J)=EL(2)                                                      WGAM0077
109   CONTINUE                                                          WGAM0078
      WRITE (NOUT,203)                                                  WGAM0079
203   FORMAT (1H0//'0BOUND  CIRCULATION'/1H0,'PSI      BVG       ELB')  WGAM0080
      WRITE (NOUT,204) (ANG(J),BVO(3,J),ELB(J),J=1,NPSI)                WGAM0081
204   FORMAT (1H ,F4.0,F12.8,F6.2)                                      WGAM0082
      GO TO 12                                                          WGAM0083
C  VORTEX SHEET WAKE MODEL                                              WGAM0084
110   CONTINUE                                                          WGAM0085
      SELB=0.0                                                          WGAM0086
      DO 112 J=1,NPSI                                                   WGAM0087
      EMS=EMU*S(J)                                                      WGAM0088
      GAM=0.0                                                           WGAM0089
      GAS=0.0                                                           WGAM0090
      DO 111 I=1,NE                                                     WGAM0091
      GA=(ETG(I)+EMS)*ALPHA(I,J)                                        WGAM0092
      GAS=GAS+DETA(I)*GA                                                WGAM0093
111   GAM=AMAX1(GAM,GA)                                                 WGAM0094
      GBV(3,J)=AP*GAM                                                   WGAM0095
      ELB(J)=2.*(1.-GAS/GAM)-EL(1)                                      WGAM0096
      SELB=SELB+ELB(J)                                                  WGAM0097
112   CONTINUE                                                          WGAM0098
      NPS=NPSI-1                                                        WGAM0099
      EL(2)=(SELB-ELB(NPSI))/NPS                                        WGAM0100
      IF (EL(2).GT.ELBM) EL(2)=ELBM                                     WGAM0101
      DO 113 J=1,NPS                                                    WGAM0102
113   GS(3,J)=GBV(3,J+1)-GBV(3,J)                                       WGAM0103
      GS(3,NPSI)=GS(3,1)                                                WGAM0104
      LST=1                                                             WGAM0105
      GO TO   1085                                                      WGAM0106
C  VORTEX LINE WAKE MODEL WITH MANY TRAILING VORTEX LINES               WGAM0107
114   DO 116  J=1,NPSI                                                  WGAM0108
```

```
      EMS=EMU*S(J)                                                      WGAM0109
      BVG(1,J)=0.0                                                      WGAM0110
      DO 115 I=1,NE                                                     WGAM0111
      BVG(I+1,J)=AP*(ETG(I)+EMS)*ALPHA(I,J)                             WGAM0112
  115 GBV(I,J)=BVG(I,J)-BVG(I+1,J)                                      WGAM0113
  116 GBV(NL,J)=BVG(NL,J)                                               WGAM0114
      NPS=NPSI-1                                                        WGAM0115
      DO 118  I=2,NL                                                    WGAM0116
      DO 117 J=1,NPS                                                    WGAM0117
      BVO(I,J)=BVG(I,J)/BOER                                            WGAM0118
  117 GS(I,J)=BVG(I,J+1)-BVG(I,J)                                       WGAM0119
      BVO(I,NPSI)=BVG(I,NPSI)/BOER                                      WGAM0120
  118 GS(I,NPSI)=GS(I,1)                                                WGAM0121
      LST=1                                                             WGAM0122
      WRITE (NOUT,200) (ETG(I),I=1,NE)                                  WGAM0123
  200 FORMAT (1H0//'0BOUND  CIRCULATION'/1H0,'ETA=',F9.3,9F11.3)        WGAM0124
      WRITE (NOUT,201)                                                  WGAM0125
  201 FORMAT (' PSI')                                                   WGAM0126
      DO 20 J=1,NPSI                                                    WGAM0127
   20 WRITE (NOUT,202) ANG(J),(BVO(I,J),I=2,NL)                         WGAM0128
  202 FORMAT (1H ,F4.0,10F11.7)                                         WGAM0129
      GO TO 12                                                          WGAM0130
C  CONSTANT GAMMA(PSI) - NO SHED WAKE                                   WGAM0131
   11 CONTINUE                                                          WGAM0132
      DO 13  J=1,NPSI                                                   WGAM0133
   13 GBV(3,J)=BOER*GZER                                                WGAM0134
      LST=2                                                             WGAM0135
      IF (IABC.EQ.5) LST=1                                              WGAM0136
      GO TO 1085                                                        WGAM0137
   12 CONTINUE                                                          WGAM0138
C  INITIALIZE DISTORTION                                                WGAM0139
       IF(I1.GT.0)  GO TO 18                                            WGAM0140
C   THIS IS A FIRST ITERATION                                           WGAM0141
C  USE RIGID WAKE FOR INITIAL ESTIMATE                                  WGAM0142
      DO 17 J=1,NPSI                                                    WGAM0143
      DO 17 K=1,NPHI                                                    WGAM0144
```

```
      D(1,J,K)=0.0                                                      WGAM0145
      D(2,J,K)=0.0                                                      WGAM0146
17    D(3,J,K)=TAMB*PSI(K)                                              WGAM0147
      GO TO 14                                                          WGAM0148
C   THIS IS AN ITERATION AFTER THE FIRST                                WGAM0149
C  READ IN RESULTS OF PREVIOUS ITERATION                                WGAM0150
18    CONTINUE                                                          WGAM0151
      DO 195 K=1,NPHI                                                   WGAM0152
      READ (NIN,19)((D(I,J,K),I=1,3),J=1,NPSI)                          WGAM0153
19    FORMAT (6E13.6)                                                   WGAM0154
195   CONTINUE                                                          WGAM0155
14    CONTINUE                                                          WGAM0156
      RETURN                                                            WGAM0157
      END                                                               WGAM0158
```

```
      SUBROUTINE DCALC                                                  DCAL0001
C  WG 71                                                                DCAL0002
C  COMPUTE D(N,J,LL) = WAKE DISTORTION                                  DCAL0003
C  P-ETA = POINT IN WAKE WHERE D(N,J,LL) IS CURRENTLY BEING COMPUTED    DCAL0004
C  N = 1,3 = INDEX FOR X,Y,Z COMPONENTS OF DISTORTION VECTOR            DCAL0005
C  J = 1,NPSI = INDEX FOR PSI = AZIMUTH ANGLE OF P-ETA                  DCAL0006
C  LL = 1,NPHI = INDEX FOR DELTA = AGE OF P-ETA                         DCAL0007
      INTEGER OKPHI,SKPHI                                               DCAL0008
      DIMENSION OLD(3)                                                  DCAL0009
      COMMON XA,YA,ZA,XB,YB,ZB,ZM,DELL,GA,GB,RHOC,RHOS,QI,QJ,QK,        DCAL0010
     1 IEG,I,J,K,KP,KPHI,KT,KDLL,L,LD,LG,LL,MGO,NGO,ND,KPO,KSTRT,       DCAL0011
     2 QMO,QMB,SQ                                                       DCAL0012
      COMMON/SSPLOT/EMU,AMBDA,ETA,C(97),S(97),PSI(97),D(3,25,97)        DCAL0013
      COMMON/SGAM/IABC,I1,IG,IS,LST,NL,NB,NPSI,NPHI,NIN,NOUT,NDI,       DCAL0014
     1 EM,DPSI,DANG,TAMB,GZER,BOER,                                     DCAL0015
     2 EL(6),ELB(25),ANG(97),ZETA(6),GBV(6,25),BVG(6,25),GS(6,25),      DCAL0016
     3 ENDM(25),NDM(25)                                                 DCAL0017
      COMMON/SQCAL/OKPHI,SKPHI,OLDM,PHI,SUMI,SUMJ,SUMK,                 DCAL0018
     1 OSUMI,OSUMJ,OSUMK,TANMU,TAMD,PZA,PZAP,AO,ETASP,ETACP,DELTA,DMS,  DCAL0019
     2 RHOK,RHOG,START,NBIGM,                                           DCAL0020
     1 AD(3),D3(3),VI(6),VJ(6),VK(6),OVI(6),OVJ(6),OVK(6),              DCAL0021
     2 DVO(3,25),DS(3,25),DSN(3,25),TOT(3,25),CUR(3,25),QBV(3),         DCAL0022
     3 KM(6,25),KTR(6,25,16),VEL(3,25,97),KPB(25)                       DCAL0023
      COMMON/SDCAL/IPR,NPSR,LDM,FWD,EWD,DSMI,DSMJ,DSMK,DSMS             DCAL0024
C  INITIALIZE                                                           DCAL0025
      QMS=QMO                                                           DCAL0026
      IEG=2                                                             DCAL0027
      KT=0                                                              DCAL0028
      MGO=1                                                             DCAL0029
      NGO=1                                                             DCAL0030
      KDLL=1                                                            DCAL0031
      LD=2                                                              DCAL0032
      LS=2                                                              DCAL0033
      LL1=2                                                             DCAL0034
      LL2=2                                                             DCAL0035
C  START BOUNDARY AGE LOOP                                              DCAL0036
```

```
23    CONTINUE                                                          DCAL0037
      LL=LL1                                                            DCAL0038
C  START AGE (DELTA) LOOP                                               DCAL0039
230   CONTINUE                                                          DCAL0040
      DELTA=PSI(LL)                                                     DCAL0041
C  START AZIMUTH ANGLE (PSI) LOOP                                       DCAL0042
      DO 403 J=1,NPSI                                                   DCAL0043
      ETASP=ETA*S(J)                                                    DCAL0044
      ETACP=ETA*C(J)                                                    DCAL0045
C  ESTIMATE DISTORTION OF P-ETA                                         DCAL0046
      D3(1)=D(1,J,LL)                                                   DCAL0047
      D3(2)=D(2,J,LL)                                                   DCAL0048
      D3(3)=D(3,J,LL)                                                   DCAL0049
      IF (MGO.LE.1) GO TO 233                                           DCAL0050
      IF (LL.LE.KDLL) GO TO 232                                         DCAL0051
C  KDLL - BOUNDARY BETWEEN CALCULATED & ASSUMED DISTORTION              DCAL0052
C  DS(N,J) = ASSUMED VALUE OF D(N,J,KDLL)                               DCAL0053
      D3(1)=D3(1)+D(1,J,KDLL)-DS(1,J)                                   DCAL0054
      D3(2)=D3(2)+D(2,J,KDLL)-DS(2,J)                                   DCAL0055
      D3(3)=D3(3)+D(3,J,KDLL)-DS(3,J)                                   DCAL0056
      IF (KDLL.EQ.LD) GO TO 233                                         DCAL0057
232   IF (LL.LE.LD) GO TO 233                                           DCAL0058
C  MODIFY D(N,J,LL) TO INCLUDE UPDATED D(N,J,LD) INFORMATION            DCAL0059
C  LD = VALUE OF LL DURING MOST RECENT COMPUTATION OF D(N,J,LL)         DCAL0060
C  DVO(N,J) = VALUE OF D(N,J,LD) BEFORE MOST RECENT UPDATE              DCAL0061
      D3(1)=D3(1)+D(1,J,LD)-DVO(1,J)                                    DCAL0062
      D3(2)=D3(2)+D(2,J,LD)-DVO(2,J)                                    DCAL0063
      D3(3)=D3(3)+D(3,J,LD)-DVO(3,J)                                    DCAL0064
233   CONTINUE                                                          DCAL0065
      QMO=QMS                                                           DCAL0066
C  CHECK FOR FIRST PASS                                                 DCAL0067
      IF (MGO.LE.1) GO TO 24                                            DCAL0068
C  SURPRESS QMO OUTPUT DURING GENERAL UPDATE                            DCAL0069
      IF (MGO.EQ.2) QMO=1.0                                             DCAL0070
235   DLDM=(LL-2)/FNDM(J)-(LL-2)/NDM(J)                                 DCAL0071
C  CHECK FOR BOUNDARY UPDATE                                            DCAL0072
```

```
      IF (OLDM.GT.0.01) GO TO 25                                        DCAL0073
C  SURPRESS QMO OUTPUT DURING BOUNDARY UPDATE                           DCAL0074
      QMO=1.0                                                           DCAL0075
24    CONTINUE                                                          DCAL0076
C  EITHER BOUNDARY UPDATE OR FIRST PASS                                 DCAL0077
C  RECOMPUTE INDUCED VELOCITY (Q) FOR ALL AGES                          DCAL0078
      TOT(1,J)=0.0                                                      DCAL0079
      TOT(2,J)=0.0                                                      DCAL0080
      TOT(3,J)=0.0                                                      DCAL0081
25    SUMI=TOT(1,J)                                                     DCAL0082
      SUMJ=TOT(2,J)                                                     DCAL0083
      SUMK=TOT(3,J)                                                     DCAL0084
C  INITIALIZE                                                           DCAL0085
      OSUMI=SUMI                                                        DCAL0086
      OSUMJ=SUMJ                                                        DCAL0087
      OSUMK=SUMK                                                        DCAL0088
      QBV(1)=0.0                                                        DCAL0089
      QBV(2)=0.0                                                        DCAL0090
      QBV(3)=0.0                                                        DCAL0091
C  COMPUTE Q                                                            DCAL0092
      CALL TRANS                                                        DCAL0093
      IF (MGO.LE.1) GO TO 96                                            DCAL0094
      IF (OLDM.GT.0.01) GO TO 96                                        DCAL0095
C  ADD IN FAR WAKE CONTRIBUTION (ALL Q.LT.DM)                           DCAL0096
      SUMI=SUMI+VEL(1,J,LL)                                             DCAL0097
      SUMJ=SUMJ+VEL(2,J,LL)                                             DCAL0098
      SUMK=SUMK+VEL(3,J,LL)                                             DCAL0099
96    CONTINUE                                                          DCAL0100
C  AVERAGE Q AT START OF LAST DELTA INCREMENT WITH Q AT END             DCAL0101
C  ADD IN Q DUE TO BOUND VORTICITY                                      DCAL0102
      CUR(1,J)=SUMI-0.5*DSMI+QBV(1)                                     DCAL0103
      CUR(2,J)=SUMJ-0.5*DSMJ+QBV(2)                                     DCAL0104
      CUR(3,J)=SUMK-0.5*DSMK+QBV(3)                                     DCAL0105
C  SAVE Q FOR POSSIBLE FUTURE USE                                       DCAL0106
      TOT(1,J)=SUMI                                                     DCAL0107
      TOT(2,J)=SUMJ                                                     DCAL0108
```

```
      TOT(3,J)=SUMK                                                     DCAL0109
C  END AZIMUTH ANGLE (PSI) LOOP                                         DCAL0110
403   CONTINUE                                                          DCAL0111
C  COMPUTE DISTORTION                                                   DCAL0112
C  AVERAGE PREVIOUS AND CURRENT COMPUTATIONS OF INCREMENTAL DISTORTION  DCAL0113
C  DURING ONE DELTA PSI STEP IN AGE USING WEIGHTING FACTOR FWD          DCAL0114
      IF (LL.EQ.2) GO TO 415                                            DCAL0115
      DO 410 N=1,3                                                      DCAL0116
      DO 410 M=1,NPSI                                                   DCAL0117
      OLD(N)=D(N,M,LL)-DVO(N,M)                                         DCAL0118
      DVO(N,M)=D(N,M,LL)                                                DCAL0119
      D(N,M,LL)=D(N,M,LL-1)+DANG*FWD*CUR(N,M)+EWD*OLD(N)                DCAL0120
410   CONTINUE                                                          DCAL0121
      GO TO 425                                                         DCAL0122
415   CONTINUE                                                          DCAL0123
      DO 420 N=1,3                                                      DCAL0124
      DO 420 M=1,NPSI                                                   DCAL0125
      DVO(N,M)=D(N,M,LL)                                                DCAL0126
      D(N,M,LL)=DANG*FWD*CUR(N,M)+EWD*DVO(N,M)                          DCAL0127
420   CONTINUE                                                          DCAL0128
425   CONTINUE                                                          DCAL0129
      LD=LL                                                             DCAL0130
C  END AGE (DELTA) LOOP                                                 DCAL0131
      IF (LL.GE.LL2) GO TO 450                                          DCAL0132
      LL=LL+1                                                           DCAL0133
      GO TO 230                                                         DCAL0134
450   CONTINUE                                                          DCAL0135
C  CHECK FOR JUST ENDED GENERAL UPDATE                                  DCAL0136
      IF (MGO.EQ.2) GO TO 470                                           DCAL0137
C  KDLL = LARGEST VALUE OF LL FOR WHICH D(N,J,LL) HAS BEEN COMPUTED     DCAL0138
      KDLL=LD                                                           DCAL0139
C  CHECK TO SEE IF CURRENT ITERATION FINNISHED                          DCAL0140
      IF (KDLL.GE.NPHI) RETURN                                          DCAL0141
C  CHECK FOR NEXT GENERAL UPDATE                                        DCAL0142
      IF (LL-LS.LT.LDM) GO TO 471                                       DCAL0143
C  START GENERAL UPDATE OF Q AND D                                      DCAL0144
```

```
C  RECOMPUTE ALL Q.GT.DM (NEAR WAKE)
      MGO=2
      LL1=2
      LL2=KDLL
      LS=LD
      IF (IPR.LT.1) GO TO 23
C  PRINT INTERMEDIATE RESULTS
      DO 4500 M=1,NPSR,IPR
      DO 4500 N=1,3
4500  WRITE (6,4501) KDLL,N,M,
     1 DS(N,M),DSN(N,M),DVO(N,M),CUR(N,M),VEL(N,M,KDLL),QBV(N),
     2 (D(N,M,K),K=2,NPHI)
4501  FORMAT (' KDLL=',I2,' N=',I1,' J=',I2,
     1 ' DS=',F7.4,' DSN=',F7.4,' DVO=',F7.4,' CUR=',F7.4,' VEL=',F7.4,
     2 ' QBV=',F9.6/
     3 (1X,18F7.4))
      GO TO 23
C  NO GENERAL UPDATE
C  INCREMENT BOUNDARY AGE
C  COMPUTE ONLY Q DUE TO NEW WAKE GENERATED BY INCREMENTING DELTA
C  ALL OTHER Q REMAINS UNCHANGED
470   CONTINUE
471   LL1=KDLL+1
      LL2=LL1
      MGO=3
      KDLS=LL1
C  UPDATE VEL, DS, DSN ARRAYS
      DO 476 N=1,3
      DO 476 J=1,NPSI
C  VEL = SUM ALL Q.LT.DM
      VEL(N,J,LL1)=VEL(N,J,KDLL)
C  DS,DSN = SAVED VALUES OF D
      DS(N,J)=DSN(N,J)
476   DSN(N,J)=D(N,J,KDLS)
      GO TO 23
      END
```

```
       SUBROUTINE TRANS                                                    TRAN0001
C  WG 71                                                                   TRAN0002
C  TRANS DOES BOOKKEEPING TO INSURE THAT ONLY Q.GT.DM (NEAR WAKE) UPDATED  TRAN0003
       INTEGER OKPHI,SKPHI                                                 TRAN0004
       COMMON XA,YA,ZA,XB,YB,ZB,ZM,DELL,GA,GB,RHOC,RHOS,QI,QJ,QK,          TRAN0005
     1 IEG,I,J,K,KP,KPHI,KT,KDLL,L,LD,LG,LL,MGO,NGO,ND,KPO,KSTRT,          TRAN0006
     2 QMO,QMB,SQ                                                          TRAN0007
       COMMON/SSPLOT/EMU,AMBDA,ETA,C(97),S(97),PSI(97),D(3,25,97)          TRAN0008
       COMMON/SGAM/IABC,I1,IG,IS,LST,NL,NB,NPSI,NPHI,NIN,NOUT,NDI,         TRAN0009
     1 EM,DPSI,DANG,TAMB,GZER,BOER,                                        TRAN0010
     2 EL(6),ELB(25),ANG(97),ZETA(6),GBV(6,25),BVG(6,25),GS(6,25),         TRAN0011
     3 ENDM(25),NDM(25)                                                    TRAN0012
       COMMON/SQCAL/OKPHI,SKPHI,OLDM,PHI,SUMI,SUMJ,SUMK,                   TRAN0013
     1 DSUMI,DSUMJ,DSUMK,TANMU,TAMD,PZA,PZAP,AO,ETASP,ETACP,DELTA,DMS,     TRAN0014
     2 RHOK,RHOG,START,NBIGM,                                              TRAN0015
     1 AD(3),D3(3),VI(6),VJ(6),VK(6),DVI(6),DVJ(6),DVK(6),                 TRAN0016
     2 DVO(3,25),DS(3,25),DSN(3,25),TOT(3,25),CUR(3,25),QBV(3),            TRAN0017
     3 KM(6,25),KTR(6,25,16),VEL(3,25,97),KPB(25)                          TRAN0018
       COMMON/SDCAL/IPR,NPSR,LDM,FWD,EWD,DSMI,DSMJ,DSMK,DSMS               TRAN0019
       COMMON/SWQC/LIB,RHOI,RHOL,RHO                                       TRAN0020
       DSMI=0.0                                                            TRAN0021
       DSMJ=0.0                                                            TRAN0022
       DSMK=0.0                                                            TRAN0023
C  P-ETA = POINT IN WAKE WHERE D(N,J,LL) IS CURRENTLY BEING COMPUTED       TRAN0024
C  POINT P-EL LOCATES AN ELEMENT IN THE WAKE WHOSE INDUCED VELOCITY        TRAN0025
C  AT POINT P-ETA IS BEING COMPUTED                                        TRAN0026
C  ETA-BLADE = BLADE WHICH TRAILED POINT P-ETA                             TRAN0027
C  EL-BLADE = BLADE WHICH TRAILED POINT P-EL                               TRAN0028
C  I = INDEX FOR ZETA = AZIMUTH ANGLE BETWEEN ETA-BLADE AND EL-BLADE       TRAN0029
C  START ZETA LOOP                                                         TRAN0030
       DO 400 I=1,NB                                                       TRAN0031
       PZA=PSI(J)+ZETA(I)                                                  TRAN0032
C  INITIALIZE AGE DELTA-L, AZIMUTH ANGLE PHI, AND TRANSITION POINT         TRAN0033
C  KPHI=1,NPSI - INDEX FOR PRINCIPLE VALUE OF PHI                          TRAN0034
C  PHI - AZIMUTH ANGLE OF POINT P-EL                                       TRAN0035
C  KP=1,NBIGM - INDEX FOR DELTA-L - AGE OF POINT P-EL                      TRAN0036
```

```
C  CHECK FOR FIRST PASS                                                     TRAN0037
      IF (MGO.LE.1)  GO TO 30                                               TRAN0038
C  CHECK FOR BOUNDARY UPDATE                                                TRAN0039
      IF (OLDM.LT.0.01) GO TO 28                                            TRAN0040
C  COMPUTE ONLY Q DUE TO NEW WAKE GENERATED BY INCREMENTING DELTA           TRAN0041
26    KT=-LL-1                                                              TRAN0042
      KP=2                                                                  TRAN0043
      PHI=DELTA+PZA-DANG                                                    TRAN0044
      KPHI=PHI/DANG+1.1                                                     TRAN0045
C  KPHI ONLY FOR PRINCIPLE VALUE OF PHI                                     TRAN0046
27    IF (KPHI.LT.NPSI) GO TO 31                                            TRAN0047
      KPHI=KPHI-NPSI+1                                                      TRAN0048
      GO TO 27                                                              TRAN0049
C  BOUNDARY UPDATE - RECOMPUTE ALL Q.GT.DM (NEAR WAKE)                      TRAN0050
C  KTR(I,J,K) = TABLE OF TRANSISTIONS BETWEEN Q.GT.DM AND Q.LT.DM           TRAN0051
C  KM(I,J) = TABLE OF MAX VALUES OF K                                       TRAN0052
28    IF (KM(I,J).EQ.0) GO TO 26                                            TRAN0053
      KP=KTR(I,J,1)+LL                                                      TRAN0054
      IF (KP.LT.2) GO TO 26                                                 TRAN0055
C  KT = NEXT TRANSISTION FROM Q.GT.DM TO Q.LT.DM                            TRAN0056
      KT=KTR(I,J,2)                                                         TRAN0057
C  K = TRANSISTION NUMBER                                                   TRAN0058
      K=2                                                                   TRAN0059
      KPHI=J+LL-KP+NDI*(I-1)                                                TRAN0060
      PHI=DANG*(KPHI-1)                                                     TRAN0061
C  KPHI ONLY FOR PRINCIPLE VALUE OF PHI                                     TRAN0062
285   IF (KPHI.GT.0) GO TO 29                                               TRAN0063
      KPHI=KPHI+NPSI-1                                                      TRAN0064
      GO TO 285                                                             TRAN0065
29    IF (KPHI.LT.NPSI) GO TO 31                                            TRAN0066
      KPHI=KPHI-NPSI+1                                                      TRAN0067
      GO TO 29                                                              TRAN0068
C  COMPUTE ALL Q FIRST TIME THROUGH                                         TRAN0069
30    KP=2+NBIGM                                                            TRAN0070
      MGO=1                                                                 TRAN0071
      K=0                                                                   TRAN0072
```

```
      KM(I,J)=0                                                         TRAN0073
      PHI=PZA-START                                                     TRAN0074
      KPHI=PZA/DANG+1.1                                                 TRAN0075
C  KPHI ONLY FOR PRINCIPLE VALUE OF PHI                                 TRAN0076
      IF (KPHI.GE.NPSI) KPHI=KPHI-NPSI+1                                TRAN0077
31    CONTINUE                                                          TRAN0078
C  SAVE INITIAL VALUE OF DELTA-L                                        TRAN0079
      KSTRT=KP                                                          TRAN0080
C  START AGE DELTA-L LOOP                                               TRAN0081
C  PZAP = RELATIVE AGE                                                  TRAN0082
32    PZAP=PZA-PHI                                                      TRAN0083
C  SET VORTEX CORE SIZE                                                 TRAN0084
C  KPB(KPHI) = INDEX FOR AGE AT WHICH VORTEX BURSTS                     TRAN0085
      RHO=RHOK                                                          TRAN0086
      IF (KP.GE.KPB(KPHI)) RHO=RHOG                                     TRAN0087
C  COMPUTE INDUCED VELOCITIES (Q)                                       TRAN0088
      CALL WQCAL                                                        TRAN0089
C  INCREMENT AGE DELTA-L AND AZIMUTH ANGLE PHI                          TRAN0090
      KP=KP-1                                                           TRAN0091
      PHI=PHI+DANG                                                      TRAN0092
      SKPHI=OKPHI                                                       TRAN0093
      OKPHI=KPHI                                                        TRAN0094
      KPHI=KPHI+1                                                       TRAN0095
C  RESET KPHI EVERY 360 DEGREES                                         TRAN0096
      IF (KPHI.EQ.NPSI) KPHI=1                                          TRAN0097
      DO 83 L=LIB,NL                                                    TRAN0098
      OVI(L)=VI(L)                                                      TRAN0099
      OVJ(L)=VJ(L)                                                      TRAN0100
      OVK(L)=VK(L)                                                      TRAN0101
83    CONTINUE                                                          TRAN0102
      DSI=SUMI-OSUMI                                                    TRAN0103
      DSJ=SUMJ-OSUMJ                                                    TRAN0104
      DSK=SUMK-OSUMK                                                    TRAN0105
C  CHECK FOR FIRST DELTA-L                                              TRAN0106
      IF (KP+1.EQ.KSTRT) GO TO 94                                       TRAN0107
C  CHECK ON TRANSISTIONS AND STORE ANY Q.LT.DM (FAR WAKE)               TRAN0108
```

```
C   CHECK FOR GENERAL UPDATE                                            TRAN0109
      IF (MGO.EQ.2) GO TO 87                                            TRAN0110
C   FIRST PASS - FIND TRANSITIONS AND STORE Q.LT.DM (FAR WAKE)          TRAN0111
      DSMS=DSI*DSI+DSJ*DSJ+DSK*DSK                                      TRAN0112
      OSUMI=SUMI                                                        TRAN0113
      OSUMJ=SUMJ                                                        TRAN0114
      OSUMK=SUMK                                                        TRAN0115
      IF (MGO.EQ.3) GO TO 91                                            TRAN0116
C   CHECK FOR CURVED TIP VORTEX SEGMENT                                 TRAN0117
      IF (NGO.EQ.2) GO TO 85                                            TRAN0118
      IF (DSMS.GT.DMS) GO TO 86                                         TRAN0119
C   Q.LT.DM                                                             TRAN0120
      IF (MGO.EQ.1) GO TO 84                                            TRAN0121
C   PREVIOUS Q.GT.DM                                                    TRAN0122
      MGO=1                                                             TRAN0123
C   INCREMENT TRANSISTION NUMBER K AND MAX K = KM                       TRAN0124
      K=K+1                                                             TRAN0125
      KM(I,J)=K                                                         TRAN0126
C   STORE TRANSISTION                                                   TRAN0127
      KTR(I,J,K)=KP-LL+1                                                TRAN0128
C   STORE Q                                                             TRAN0129
84    VEL(1,J,LL)=VEL(1,J,LL)+DSI                                       TRAN0130
      VEL(2,J,LL)=VEL(2,J,LL)+DSJ                                       TRAN0131
      VEL(3,J,LL)=VEL(3,J,LL)+DSK                                       TRAN0132
      GO TO 94                                                          TRAN0133
85    NGO=1                                                             TRAN0134
C   Q.GT.DM                                                             TRAN0135
86    IF (MGO.EQ.0) GO TO 94                                            TRAN0136
C   PREVIOUS Q.LT.DM                                                    TRAN0137
      MGO=0                                                             TRAN0138
      K=K+1                                                             TRAN0139
      KM(I,J)=K                                                         TRAN0140
      KTR(I,J,K)=KP-LL+2                                                TRAN0141
      GO TO 94                                                          TRAN0142
87    CONTINUE                                                          TRAN0143
C   SECOND PASS (OR LATER) - CHECK FOR TRANSISTION                      TRAN0144
```

```
      IF (KT.LT.KP-LL) GO TO 94                                         TRAN0145
C  END OF ONE Q.GT.DM SEGMENT LOOK FOR NEXT ONE                         TRAN0146
      KPD=KP                                                            TRAN0147
      K=K+2                                                             TRAN0148
C  CHECK FOR ANY MORE WAKE WITH Q.GT.DM                                 TRAN0149
      IF (K.GT.KM(I,J)) GO TO 90                                        TRAN0150
C  FOUND ANOTHER TRANSISTION                                            TRAN0151
      KT=KTR(I,J,K)                                                     TRAN0152
88    KP=KTR(I,J,K-1)+LL                                                TRAN0153
      KSTRT=KP                                                          TRAN0154
      KPD=KPD-KP                                                        TRAN0155
      PHI=PHI+DANG*KPD                                                  TRAN0156
      KPHI=KPHI+KPD                                                     TRAN0157
89    IF (KPHI.LT.NPSI) GO TO 94                                        TRAN0158
      KPHI=KPHI-NPSI+1                                                  TRAN0159
      GO TO 89                                                          TRAN0160
90    IF (K.GE.KM(I,J)+2) GO TO 905                                     TRAN0161
      KT=-LL-1                                                          TRAN0162
      GO TO 88                                                          TRAN0163
C  ALWAYS RECOMPUTE KP=1 TO GET LATEST BOUND VORTICITY CONTRIBUTION TO Q TRAN0164
905   CONTINUE                                                          TRAN0165
      IF (KP.EQ.0) GO TO 95                                             TRAN0166
      KT=-LL-1                                                          TRAN0167
      KP=2                                                              TRAN0168
      KSTRT=2                                                           TRAN0169
      PHI=DELTA+PZA-DANG                                                TRAN0170
      KPHI=PHI/DANG+1.1                                                 TRAN0171
      GO TO 89                                                          TRAN0172
91    CONTINUE                                                          TRAN0173
      IF (KP.GT.1) GO TO 87                                             TRAN0174
      ODKM=KM(I,J)/2.-KM(I,J)/2                                         TRAN0175
      IF (ODKM.GT.0.1) GO TO 92                                         TRAN0176
C  PREVIOUS Q.LT.DM                                                     TRAN0177
      IF (DSMS.LT.DMS) GO TO 93                                         TRAN0178
C  Q.GT.DM                                                              TRAN0179
      IF (KM(I,J).GE.16) GO TO 490                                      TRAN0180
```

```
      KM(I,J)=KM(I,J)+1                                                 TRAN0181
      KTR(I,J,KM(I,J))=KP-LL+2                                          TRAN0182
      GO TO 94                                                          TRAN0183
C  PREVIOUS Q.GT.DM                                                     TRAN0184
92    IF (DSMS.GT.DMS) GO TO 94                                         TRAN0185
C  Q.LT.DM                                                              TRAN0186
      KM(I,J)=KM(I,J)+1                                                 TRAN0187
      KTR(I,J,KM(I,J))=KP-LL+1                                          TRAN0188
93    VEL(1,J,LL)=VEL(1,J,LL)+DSI                                       TRAN0189
      VEL(2,J,LL)=VEL(2,J,LL)+DSJ                                       TRAN0190
      VEL(3,J,LL)=VEL(3,J,LL)+DSK                                       TRAN0191
94    CONTINUE                                                          TRAN0192
C  END AGE DELTA-L LOOP                                                 TRAN0193
      IF (KP.GE.1) GO TO 32                                             TRAN0194
95    CONTINUE                                                          TRAN0195
      DSMI=DSMI+DSI                                                     TRAN0196
      DSMJ=DSMJ+DSJ                                                     TRAN0197
      DSMK=DSMK+DSK                                                     TRAN0198
C  END ZETA LOOP                                                        TRAN0199
400   CONTINUE                                                          TRAN0200
      RETURN                                                            TRAN0201
C  TOO MANY TRANSISTION POINTS                                          TRAN0202
490   CONTINUE                                                          TRAN0203
      WRITE (NOUT,491)                                                  TRAN0204
491   FORMAT ('1TOO  MANY  TRANSISTION  POINTS')                        TRAN0205
      DO 492 J=1,NPSI                                                   TRAN0206
      WRITE (NOUT,453) ANG(J)                                           TRAN0207
      DO 492 I=1,NB                                                     TRAN0208
      KMIJ=KM(I,J)                                                      TRAN0209
      WRITE (NOUT,454) (KTR(I,J,K),K=1,KMIJ)                            TRAN0210
492   CONTINUE                                                          TRAN0211
453   FORMAT (' PSI=',F4.0)                                             TRAN0212
454   FORMAT (' KTR=',16I4)                                             TRAN0213
      STOP                                                              TRAN0214
      END                                                               TRAN0215
```

```
      SUBROUTINE WQCAL                                                  WQCL0001
C  WG 71                                                                WQCL0002
C  COMPUTE INDUCED VELOCITIES (Q)                                       WQCL0003
       INTEGER OKPHI,SKPHI                                              WQCL0004
      DIMENSION SVI(6),SVJ(6),SVK(6)                                    WQCL0005
      COMMON XA,YA,ZA,XB,YB,ZB,ZM,DELL,GA,GB,RHOC,RHOS,QI,QJ,QK,        WQCL0006
     1 IFG,I,J,K,KP,KPHI,KT,KDLL,L,LD,LG,LL,MGC,NGO,ND,KPC,KSTRT,       WQCL0007
     2 QMO,QMB,SQ                                                       WQCL0008
      COMMON/SSPLOT/EMU,AMBDA,ETA,C(97),S(97),PSI(97),D(3,25,97)        WQCL0009
      COMMON/SGAM/IABC,I1,IG,IS,LST,NL,NB,NPSI,NPHI,NIN,NOUT,NDI,       WQCL0010
     1 EM,DPSI,DANG,TAMB,GZER,BOER,                                     WQCL0011
     2 EL(6),ELB(25),ANG(97),ZETA(6),GBV(6,25),BVG(6,25),GS(6,25),      WQCL0012
     3 ENDM(25),NDM(25)                                                 WQCL0013
      COMMON/SQCAL/OKPHI,SKPHI,OLDM,PHI,SUMI,SUMJ,SUMK,                 WQCL0014
     1 OSUMI,OSUMJ,OSUMK,TANMU,TAMD,PZA,PZAP,AO,ETASP,ETACP,DELTA,DMS,  WQCL0015
     2 RHOK,RHOG,START,NBIGM,                                           WQCL0016
     1 AD(3),D3(3),VI(6),VJ(6),VK(6),OVI(6),OVJ(6),OVK(6),              WQCL0017
     2 DVO(3,25),DS(3,25),DSN(3,25),TOT(3,25),CUR(3,25),QBV(3),         WQCL0018
     3 KM(6,25),KTR(6,25,16),VEL(3,25,97),KPB(25)                       WQCL0019
      COMMON/SWQC/LIB,RHOI,RHOL,RHO                                     WQCL0020
C  L = INDEX FOR EL = RADIAL STATION ON EL-BLADE WHICH TRAILED P-EL     WQCL0021
C  START L LOOP                                                         WQCL0022
      DO 82 L=LIB,NL                                                    WQCL0023
      LG=L-1                                                            WQCL0024
C   NOW DEFINE CURRENT VECTORS                                          WQCL0025
      IF(L.EQ.ND)   GO TO 41                                            WQCL0026
C  RIGID WAKE                                                           WQCL0027
      AD(1)=-D3(1)                                                      WQCL0028
      AD(2)=-D3(2)                                                      WQCL0029
      AD(3)=TAMB*(PZAP+DELTA)-D3(3)                                     WQCL0030
      GO TO 515                                                         WQCL0031
C  DISTORTED WAKE                                                       WQCL0032
41    IF (MGO.LE.1) GO TO 511                                           WQCL0033
C  KDLL = LARGEST VALUE OF LL FOR WHICH D(N,J,LL) COMPUTED SO FAR       WQCL0034
      IF(KP.GT.KDLL)  GO TO 42                                          WQCL0035
      AD(1)= D(1,KPHI,KP)-D3(1)                                         WQCL0036
```

```
      AD(2)= D(2,KPHI,KP)-D3(2)                                               WQCL0037
      AD(3)= D(3,KPHI,KP)-D3(3)                                               WQCL0038
      GO TO 51                                                                WQCL0039
   42 IF(KP.GT.NPHI)   GO TO 44                                               WQCL0040
C  USE COMPUTED VALUES OF D FOR DISTORTION ACCUMULATED UP TO AGE KDLL         WQCL0041
C  USE INITIAL ESTIMATE OF D FOR DISTORTION BEYOND AGE KDLL                   WQCL0042
C  DS(N,J) = D(N,J,KDLL) BEFORE LATEST UPDATE                                 WQCL0043
      AD(1)=D(1,KPHI,KP)-DS(1,KPHI)+D(1,KPHI,KDLL)-D3(1)                      WQCL0044
      AD(2)=D(2,KPHI,KP)-DS(2,KPHI)+D(2,KPHI,KDLL)-D3(2)                      WQCL0045
      AD(3)=D(3,KPHI,KP)-DS(3,KPHI)+D(3,KPHI,KDLL)-D3(3)                      WQCL0046
      GO TO 51                                                                WQCL0047
C  EXTRAPOLATE DISTORTION FOR AGES BEYOND NPHI (LIMIT OF D TABLE)             WQCL0048
   44 AD(1)=D(1,KPHI,NPHI)-DS(1,KPHI)+D(1,KPHI,KDLL)-D3(1)                    WQCL0049
      AD(2)=D(2,KPHI,NPHI)-DS(2,KPHI)+D(2,KPHI,KDLL)-D3(2)                    WQCL0050
      AD(3)=D(3,KPHI,NPHI)-DS(3,KPHI)+D(3,KPHI,KDLL)-D3(3)+FLOAT(KP-          WQCL0051
     1  NPHI)*TAMD                                                            WQCL0052
   51 CONTINUE                                                                WQCL0053
      IF (KDLL.EQ.LD) GO TO 515                                               WQCL0054
C  LD = VALUE OF LL DURING MOST RECENT D(N,J,LL) COMPUTATION                  WQCL0055
      IF (KP.LE.LD.OR.KP.EQ.2) GO TO 515                                      WQCL0056
C  USE RECENTLY COMPUTED VALUES OF D FOR DISTORTION ACCUMULATED UP TO AGE LD  WQCL0057
C  USE OLDER INFORMATION FOR DISTORTION BEYOND AGE LD                         WQCL0058
      AD(1)=AD(1)+D(1,KPHI,LD)-DVO(1,KPHI)                                    WQCL0059
      AD(2)=AD(2)+D(2,KPHI,LD)-DVO(2,KPHI)                                    WQCL0060
      AD(3)=AD(3)+D(3,KPHI,LD)-DVO(3,KPHI)                                    WQCL0061
      GO TO 515                                                               WQCL0062
C  FIRST PASS - NO COMPUTED DISTORTION YET                                    WQCL0063
C  TEST FOR DISTORTION EXTRAPOLATION                                          WQCL0064
  511 IF (KP.LE.NPHI) GO TO 512                                               WQCL0065
C  EXTRAPOLATE DISTORTION                                                     WQCL0066
      AD(1)=D(1,KPHI,NPHI)-D3(1)                                              WQCL0067
      AD(2)=D(2,KPHI,NPHI)-D3(2)                                              WQCL0068
      AD(3)=D(3,KPHI,NPHI)-D3(3)+FLOAT(KP-NPHI)*TAMD                          WQCL0069
      GO TO 515                                                               WQCL0070
  512 AD(1)= D(1,KPHI,KP)-D3(1)                                               WQCL0071
      AD(2)= D(2,KPHI,KP)-D3(2)                                               WQCL0072
```

```
      AD(3)= D(3,KPHI,KP)-D3(3)                                         WQCL0073
 515  CONTINUE                                                          WQCL0074
C  COMPUTE COMPONENTS OF VECTOR FROM P-ETA TO P-EL                      WQCL0075
      VI(L)=EL(L)*S(KPHI)-ETASP  +AD(1)                                 WQCL0076
      VJ(L)=EL(L)*C(KPHI)-ETACP+EMU*PZAP+AD(2)                          WQCL0077
      VK(L)=TANMU*PZAP-AO*(EL(L)-ETA) +AD(3)                            WQCL0078
 52   CONTINUE                                                          WQCL0079
C   START LOGICAL SEPARATION TO FIND METHOD FOR Q                       WQCL0080
      IF (L.LT.LST) GO TO 81                                            WQCL0081
      IF (KP.EQ.KSTRT) GO TO 81                                         WQCL0082
      IF (IABC.GT.3) GO TO 65                                           WQCL0083
C  VORTEX LINE WAKE MODEL                                               WQCL0084
      IF (I.GT.1) GO TO 60                                              WQCL0085
      IF (LL.NE.KP) GO TO 57                                            WQCL0086
      IF (L.EQ.LST) GO TO 62                                            WQCL0087
 53   IF (L.NE.ND) GO TO 56                                             WQCL0088
C  FIRST HALF OF CURVED TIP VL                                          WQCL0089
C  SAVE VECTORS FOR LATER USE                                           WQCL0090
 54   SVI(L)=OVI(L)                                                     WQCL0091
      SVJ(L)=OVJ(L)                                                     WQCL0092
      SVK(L)=OVK(L)                                                     WQCL0093
      NGO=2                                                             WQCL0094
      GO TO 81                                                          WQCL0095
C  CONSTANT GAMMA TRAILING VL                                           WQCL0096
 56   GA=GBV(L,KPHI)                                                    WQCL0097
      XA=VI(L)                                                          WQCL0098
      YA=VJ(L)                                                          WQCL0099
      ZA=VK(L)                                                          WQCL0100
      XB=OVI(L)                                                         WQCL0101
      YB=OVJ(L)                                                         WQCL0102
      ZB=OVK(L)                                                         WQCL0103
      IF (L.EQ.ND) GO TO 563                                            WQCL0104
      IF (IABC.GT.0) GO TO 562                                          WQCL0105
C  INBOARD TRAILING VS                                                  WQCL0106
      DELL=EL(2)-EL(1)                                                  WQCL0107
      ZM=0.5*(VK(1)+OVK(1))                                             WQCL0108
```

```
      CALL QVS                                                          WQCL0109
      GO TO 565                                                         WQCL0110
C  INBOARD TRAILING VL                                                  WQCL0111
562   RHOC=RHOI                                                         WQCL0112
      GO TO 564                                                         WQCL0113
C  STRAIGHT TIP VL                                                      WQCL0114
563   RHOC=RHO                                                          WQCL0115
564   RHOS=RHOC*RHOC                                                    WQCL0116
      CALL QSVL                                                         WQCL0117
565   SUMI=SUMI+QI                                                      WQCL0118
      SUMJ=SUMJ+QJ                                                      WQCL0119
      SUMK=SUMK+QK                                                      WQCL0120
      GO TO 81                                                          WQCL0121
57    IF (L.EQ.LST) GO TO 63                                            WQCL0122
      IF (KP.GT.1) GO TO 59                                             WQCL0123
C  CONSTANT GAMMA BOUND VL                                              WQCL0124
572   GA=BVG(L,KPHI)                                                    WQCL0125
      XA=VI(LG)                                                         WQCL0126
      YA=VJ(LG)                                                         WQCL0127
      ZA=VK(LG)                                                         WQCL0128
      XB=VI(L)                                                          WQCL0129
      YB=VJ(L)                                                          WQCL0130
      ZB=VK(L)                                                          WQCL0131
      RHOC=RHOL                                                         WQCL0132
      RHOS=RHOC*RHOC                                                    WQCL0133
      CALL QSVL                                                         WQCL0134
      IF (SQ.LT.QMB) GO TO 578                                          WQCL0135
C  AVERAGE QBV AS BLADE SWEEPS THRU DEPSI                               WQCL0136
      ZM=0.50*(OVK(LG)+OVK(L))                                          WQCL0137
      L=L-1                                                             WQCL0138
      DELI=SQRT((VI(L)-OVI(L))**2+(VJ(L)-OVJ(L))**2+(VK(L)-OVK(L))**2)  WQCL0139
      L=L+1                                                             WQCL0140
      DELO=SQRT((VI(L)-OVI(L))**2+(VJ(L)-OVJ(L))**2+(VK(L)-OVK(L))**2)  WQCL0141
      DELL=0.5*(DELI+DELO)                                              WQCL0142
      CALL QVS                                                          WQCL0143
578    CONTINUE                                                         WQCL0144
```

```
      QBV(1)=QBV(1)+QI                                                  WQCL0145
      QBV(2)=QBV(2)+QJ                                                  WQCL0146
      QBV(3)=QBV(3)+QK                                                  WQCL0147
58    IF (I.GT.1) GO TO 56                                              WQCL0148
581   IF (LL-1.NE.KP) GO TO 56                                          WQCL0149
      IF (L.NE.ND) GO TO 56                                             WQCL0150
C  CURVED TIP VL                                                        WQCL0151
C  COVERS TWO VL SEGMENTS (KP=LL,LL-1)                                  WQCL0152
      GA=GBV(L,KPHI)                                                    WQCL0153
      GB=GBV(L,JKPHI)                                                   WQCL0154
      XA=VI(L)-OVI(L)                                                   WQCL0155
      YA=VJ(L)-OVJ(L)                                                   WQCL0156
      ZA=VK(L)-OVK(L)                                                   WQCL0157
      XB=SVI(L)-OVI(L)                                                  WQCL0158
      YB=SVJ(L)-OVJ(L)                                                  WQCL0159
      ZB=SVK(L)-OVK(L)                                                  WQCL0160
      RHOC=RHO                                                          WQCL0161
      RHOS=RHOC*RHOC                                                    WQCL0162
      CALL QCVL                                                         WQCL0163
      GO TO 565                                                         WQCL0164
C  SHED WAKE                                                            WQCL0165
59    IF (IS.EQ.2) GO TO 58                                             WQCL0166
591   GA=GS(L,KPHI)                                                     WQCL0167
      XA=VI(L)                                                          WQCL0168
      YA=VJ(L)                                                          WQCL0169
      ZA=VK(L)                                                          WQCL0170
      RHOC=RHOL                                                         WQCL0171
      RHOS=RHOC*RHOC                                                    WQCL0172
      IF (IABC.GT.0) GO TO 593                                          WQCL0173
C  LINEAR GAMMA SHED VL                                                 WQCL0174
      XB=VI(1)                                                          WQCL0175
      YB=VJ(1)                                                          WQCL0176
      ZB=VK(1)                                                          WQCL0177
      GB=0.0                                                            WQCL0178
      IEG=1                                                             WQCL0179
      CALL QSVL                                                         WQCL0180
```

```
      IEG=2                                                             WQCL0181
      GO TO 595                                                         WQCL0182
C  CONSTANT GAMMA SHED VL                                               WQCL0183
593   CONTINUE                                                          WQCL0184
      XB=VI(LG)                                                         WQCL0185
      YB=VJ(LG)                                                         WQCL0186
      ZB=VK(LG)                                                         WQCL0187
      CALL QSVL                                                         WQCL0188
595   CONTINUE                                                          WQCL0189
      SUMI=SUMI+QI                                                      WQCL0190
      SUMJ=SUMJ+QJ                                                      WQCL0191
      SUMK=SUMK+QK                                                      WQCL0192
      GO TO 58                                                          WQCL0193
60    IF (L.EQ.LST) GO TO 64                                            WQCL0194
      IF (KP.EQ.1) GO TO 572                                            WQCL0195
      IF (IS.NE.2) GO TO 591                                            WQCL0196
      GO TO 56                                                          WQCL0197
62    IF (IABC.EQ.1) GO TO 54                                           WQCL0198
      GO TO 53                                                          WQCL0199
63    IF (IABC.EQ.1) GO TO 81                                           WQCL0200
      GO TO 581                                                         WQCL0201
64    IF (IABC.EQ.1) GO TO 81                                           WQCL0202
      GO TO 56                                                          WQCL0203
C  VORTEX SHEET WAKE MODEL                                              WQCL0204
65    CALL VSCAL                                                        WQCL0205
81    CONTINUE                                                          WQCL0206
C  END L LOOP                                                           WQCL0207
82    CONTINUE                                                          WQCL0208
      RETURN                                                            WQCL0209
      END                                                               WQCL0210
```

```
      SUBROUTINE VSCAL
C  WG 71
C  VORTEX SHEET WAKE MODEL
      INTEGER OKPHI,SKPHI
      DIMENSION SVI(6),SVJ(6),SVK(6)
      COMMON XA,YA,ZA,XB,YB,ZB,ZM,DELL,GA,GB,RHOC,RHOS,QI,QJ,QK,
     1 IEG,I,J,K,KP,KPHI,KT,KDLL,L,LD,LG,LL,MGO,NGO,ND,KPO,KSTRT,
     2 QMO,QMB,SQ
      COMMON/SGAM/IABC,I1,IG,IS,LST,NL,NB,NPSI,NPHI,NIN,NOUT,NDI,
     1 EM,DPSI,DANG,TAMB,GZER,BOER,
     2 EL(6),ELB(25),ANG(97),ZETA(6),GBV(6,25),BVG(6,25),GS(6,25),
     3 ENDM(25),NDM(25)
      COMMON/SQCAL/OKPHI,SKPHI,OLDM,PHI,SUMI,SUMJ,SUMK,
     1 OSUMI,OSUMJ,OSUMK,TANMU,TAMD,PZA,PZAP,AO,ETASP,ETACP,DELTA,DMS,
     2 RHOK,RHOG,START,NBIGM,
     1 AD(3),D3(3),VI(6),VJ(6),VK(6),OVI(6),OVJ(6),OVK(6),
     2 DVC(3,25),DS(3,25),DSN(3,25),TOT(3,25),CUR(3,25),QBV(3),
     3 KM(6,25),KTR(6,25,16),VEL(3,25,97),KPB(25)
65    CONTINUE
      IF (LL.NE.KP) GO TO 70
66    SVI(L)=OVI(L)
      SVJ(L)=OVJ(L)
      SVK(L)=OVK(L)
      NGO=2
      IF (L.EQ.1) GO TO 81
      IF (I.GT.1) GO TO 67
      IF (L.NE.ND) GO TO 67
      GO TO 81
67    IF (IABC.EQ.4) GO TO 68
      IF (L.EQ.2) GO TO 69
C  LINEAR GAMMA TRAILING VL
68    IEG=1
      GA=GBV(L,KPHI)
      GB=GBV(L,OKPHI)
      XA=VI(L)
      YA=VJ(L)
```

```
      ZA=VK(L)                                                          VSCL0037
      XB=OVI(L)                                                         VSCL0038
      YB=OVJ(L)                                                         VSCL0039
      ZB=OVK(L)                                                         VSCL0040
      CALL QSVL                                                         VSCL0041
 685  SUMI=SUMI+QI                                                      VSCL0042
      SUMJ=SUMJ+QJ                                                      VSCL0043
      SUMK=SUMK+QK                                                      VSCL0044
      GO TO 81                                                          VSCL0045
C  LINEAR GAMMA TRAILING VS                                             VSCL0046
 69   IEG=1                                                             VSCL0047
      GA=GBV(2,KPHI)                                                    VSCL0048
      GB=GBV(2,OKPHI)                                                   VSCL0049
      DELL=EL(2)-EL(1)                                                  VSCL0050
      XB=OVI(2)                                                         VSCL0051
      YB=OVJ(2)                                                         VSCL0052
      ZB=OVK(2)                                                         VSCL0053
      ZM=0.5*(VK(1)+OVK(1))                                             VSCL0054
      XA=VI(2)                                                          VSCL0055
      YA=VJ(2)                                                          VSCL0056
      ZA=VK(2)                                                          VSCL0057
      CALL QVS                                                          VSCL0058
      GO TO 685                                                         VSCL0059
 70   IF (IABC.EQ.5) GO TO 76                                           VSCL0060
      IF (L.EQ.LST) GO TO 75                                            VSCL0061
 705  IF (KP.GT.1) GO TO 72                                             VSCL0062
C  CONSTANT GAMMA BOUND VL                                              VSCL0063
 71   IEG=2                                                             VSCL0064
      GA=BVG(L,KPHI)                                                    VSCL0065
      XA=VI(LG)                                                         VSCL0066
      YA=VJ(LG)                                                         VSCL0067
      ZA=VK(LG)                                                         VSCL0068
      XB=VI(L)                                                          VSCL0069
      YB=VJ(L)                                                          VSCL0070
      ZB=VK(L)                                                          VSCL0071
      CALL QSVL                                                         VSCL0072
```

```
      IF (SQ.LT.QMB) GO TO 715                                          VSCL0073
C  AVERAGE QBV AS BLADE SWEEPS THRU DEPSI                               VSCL0074
      ZM=0.50*(OVK(LG)+OVK(L))                                          VSCL0075
      L=L-1                                                             VSCL0076
      DELI=SQRT((VI(L)-OVI(L))**2+(VJ(L)-OVJ(L))**2+(VK(L)-OVK(L))**2)  VSCL0077
      L=L+1                                                             VSCL0078
      DELO=SQRT((VI(L)-OVI(L))**2+(VJ(L)-OVJ(L))**2+(VK(L)-OVK(L))**2)  VSCL0079
      DELL=0.5*(DELI+DELO)                                              VSCL0080
      CALL QVS                                                          VSCL0081
715    CONTINUE                                                         VSCL0082
      QBV(1)=QBV(1)+QI                                                  VSCL0083
      QBV(2)=QBV(2)+QJ                                                  VSCL0084
      QBV(3)=QBV(3)+QK                                                  VSCL0085
72    IF (LL-1.NE.KP) GO TO 78                                          VSCL0086
C  CONSTANT GAMMA NEAR SHED VS                                          VSCL0087
73    IEG=2                                                             VSCL0088
      GA=0.5*(GS(3,OKPHI)+GS(3,SKPHI))                                  VSCL0089
      ZM=0.5*(VK(2)+VK(3))                                              VSCL0090
      DELO= SQRT((VI(3)-SVI(3))**2+(VJ(3)-SVJ(3))**2+(VK(3)-SVK(3))**2) VSCL0091
      DELI=SQRT((VI(2)-SVI(2))**2+(VJ( 2)-SVJ( 2))**2+( VK( 2)-SVK(2))**2) VSCL0092
      DELL=.5*(DELO+DELI)                                               VSCL0093
      XA=SVI(3)                                                         VSCL0094
      YA=SVJ(3)                                                         VSCL0095
      ZA=SVK(3)                                                         VSCL0096
      XB=SVI(2)                                                         VSCL0097
      YB=SVJ(2)                                                         VSCL0098
      ZB=SVK(2)                                                         VSCL0099
      CALL QVS                                                          VSCL0100
      SUMI=SUMI+QI                                                      VSCL0101
      SUMJ=SUMJ+QJ                                                      VSCL0102
      SUMK=SUMK+QK                                                      VSCL0103
74    IF (I.GT.1) GO TO 68                                              VSCL0104
      IF (L.NE.ND) GO TO 68                                             VSCL0105
C  CURVED TRAILING VL                                                   VSCL0106
      GA=GBV(L,KPHI)                                                    VSCL0107
      GB=GBV(L,OKPHI)                                                   VSCL0108
```

```
      XA=VI(L)-OVI(L)                                                   VSCL0109
      YA=VJ(L)-OVJ(L)                                                   VSCL0110
      ZA=VK(L)-OVK(L)                                                   VSCL0111
      XB=SVI(L)-OVI(L)                                                  VSCL0112
      YB=SVJ(L)-OVJ(L)                                                  VSCL0113
      ZB=SVK(L)-OVK(L)                                                  VSCL0114
      CALL QCVL                                                         VSCL0115
      GO TO 685                                                         VSCL0116
75    IF (LL-1.EQ.KP) GO TO 74                                          VSCL0117
      GO TO 68                                                          VSCL0118
76    IF (L.EQ.3) GO TO 705                                             VSCL0119
      IF (L.EQ.1) GO TO 81                                              VSCL0120
761   IF (KP.GT.1) GO TO 77                                             VSCL0121
C  LINEAR GAMMA BOUND VL                                                VSCL0122
      IEG=1                                                             VSCL0123
      GA=BVG(3,KPHI)                                                    VSCL0124
      GB=0.D0                                                           VSCL0125
      XA=VI(2)                                                          VSCL0126
      YA=VJ(2)                                                          VSCL0127
      ZA=VK(2)                                                          VSCL0128
      XB=VI(1)                                                          VSCL0129
      YB=VJ(1)                                                          VSCL0130
      ZB=VK(1)                                                          VSCL0131
      CALL QSVL                                                         VSCL0132
      IF (SQ.LT.QMB) GO TO 765                                          VSCL0133
C  AVERAGE QBV AS BLADE SWEEPS THRU DEPSI                               VSCL0134
      ZM=0.50*(OVK(LG)+OVK(L))                                          VSCL0135
      L=L-1                                                             VSCL0136
      DELI=SQRT((VI(L)-OVI(L))**2+(VJ(L)-OVJ(L))**2+(VK(L)-OVK(L))**2)  VSCL0137
      L=L+1                                                             VSCL0138
      DELO=SQRT((VI(L)-OVI(L))**2+(VJ(L)-OVJ(L))**2+(VK(L)-OVK(L))**2)  VSCL0139
      DELL=0.5*(DELI+DELO)                                              VSCL0140
      CALL QVS                                                          VSCL0141
765    CONTINUE                                                         VSCL0142
      QBV(1)=QBV(1)+QI                                                  VSCL0143
      QBV(2)=QBV(2)+QJ                                                  VSCL0144
```

```
      QBV(3)=QBV(3)+QK                                                  VSCL0145
77    IF (LL-1.EQ.KP) GO TO 794                                         VSCL0146
C  LINEAR GAMMA SHED VS                                                 VSCL0147
775   IEG=1                                                             VSCL0148
      GA=GS(3,OKPHI)                                                    VSCL0149
      GB=0.0                                                            VSCL0150
      ZM=0.5*(VK(1)+VK(2))                                              VSCL0151
      DELO=SQRT((VI(2)-OVI(2))**2+(VJ(2)-OVJ(2))**2+(VK(2)-OVK(2))**2)  VSCL0152
      DELI=SQRT((VI(1)-OVI(1))**2+(VJ(1)-OVJ(1))**2+(VK(1)-OVK(1))**2)  VSCL0153
      DELL=.5*(DELO+DELI)                                               VSCL0154
      XA=OVI(2)                                                         VSCL0155
      YA=OVJ(2)                                                         VSCL0156
      ZA=OVK(2)                                                         VSCL0157
      XB=OVI(1)                                                         VSCL0158
      YB=OVJ(1)                                                         VSCL0159
      ZB=OVK(1)                                                         VSCL0160
      CALL QVS                                                          VSCL0161
777   SUMI=SUMI+QI                                                      VSCL0162
      SUMJ=SUMJ+QJ                                                      VSCL0163
      SUMK=SUMK+QK                                                      VSCL0164
      GO TO 69                                                          VSCL0165
C  CONSTANT GAMMA SHED VS                                               VSCL0166
   78 IEG=2                                                             VSCL0167
      GA=GS(3,OKPHI)                                                    VSCL0168
      ZM=0.5*(VK(2)+VK(3))                                              VSCL0169
      DELO=SQRT((VI(3)-OVI(3))**2+(VJ(3)-OVJ(3))**2+(VK(3)-OVK(3))**2)  VSCL0170
      DELI=SQRT((VI(2)-OVI(2))**2+(VJ(2)-OVJ(2))**2+(VK(2)-OVK(2))**2)  VSCL0171
      DELL=.5*(DELO+DELI)                                               VSCL0172
      XA=OVI(3)                                                         VSCL0173
      YA=OVJ(3)                                                         VSCL0174
      ZA=OVK(3)                                                         VSCL0175
      XB=OVI(2)                                                         VSCL0176
      YB=OVJ(2)                                                         VSCL0177
      ZB=OVK(2)                                                         VSCL0178
      CALL QVS                                                          VSCL0179
      SUMI=SUMI+QI                                                      VSCL0180
```

```
      SUMJ=SUMJ+QJ                                                      VSCL0181
      SUMK=SUMK+QK                                                      VSCL0182
      GO TO 68                                                          VSCL0183
C  LINEAR GAMMA NEAR SHED VS                                            VSCL0184
794   IEG=1                                                             VSCL0185
      GA=0.5*(GS(3,OKPHI)+GS(3,SKPHI))                                  VSCL0186
      ZM=0.5*(VK(1)+VK(2))                                              VSCL0187
      DELO=SQRT((VI(2)-SVI(2))**2+(VJ(2)-SVJ(2))**2+(VK(2)-SVK(2))**2)  VSCL0188
      DELI=SQRT((VI(1)-SVI(1))**2+(VJ(1)-SVJ(1))**2+(VK(1)-SVK(1))**2)  VSCL0189
      DELL=.5*(DELO+DELI)                                               VSCL0190
      XA=SVI(2)                                                         VSCL0191
      YA=SVJ(2)                                                         VSCL0192
      ZA=SVK(2)                                                         VSCL0193
      XB=SVI(1)                                                         VSCL0194
      YB=SVJ(1)                                                         VSCL0195
      ZB=SVK(1)                                                         VSCL0196
      CALL QVS                                                          VSCL0197
      GO TO 777                                                         VSCL0198
81    RETURN                                                            VSCL0199
      END                                                               VSCL0200
```

```
      SUBROUTINE QSVL                                                   QSVL0001
C     WG 71                                                             QSVL0002
C     STRAIGHT VORTEX LINE SEGMENT INDUCED VELOCITY                     QSVL0003
      COMMON XA,YA,ZA,XB,YB,ZB,ZM,DELL,GA,GB,RHOC,RHOS,QI,QJ,QK,        QSVL0004
     1 IEG,I,J,K,KP,KPHI,KT,KDLL,L,LD,LG,LL,MGO,NGO,ND,KPO,KSTRT,       QSVL0005
     2 QMO,QMB,SQ                                                       QSVL0006
   30 ADOTB=XA*XB+YA*YB+ZA*ZB                                           QSVL0007
      AC=XA*XA+YA*YA+ZA*ZA                                              QSVL0008
      BC=XB*XB+YB*YB+ZB*ZB                                              QSVL0009
      AL=SQRT(AC)                                                       QSVL0010
      BL=SQRT(BC)                                                       QSVL0011
      CLS=AC+BC-2.*ADOTB                                                QSVL0012
      AXBI=YA*ZB-YB*ZA                                                  QSVL0013
      AXBJ=ZA*XB-ZB*XA                                                  QSVL0014
      AXBK=XA*YB-XB*YA                                                  QSVL0015
      AXBS=AXBI**2+AXBJ**2+AXBK**2                                      QSVL0016
      HS=AXBS/CLS                                                       QSVL0017
      QF=HS/(RHOS+HS)                                                   QSVL0018
5     Q=QF*GA*(AL+BL)*(1.0-ADOTB/(AL*BL))/AXBS                          QSVL0019
      QI=Q*AXBI                                                         QSVL0020
      QJ=Q*AXBJ                                                         QSVL0021
      QK=Q*AXBK                                                         QSVL0022
      IF (IEG.EQ.2) GO TO 40                                            QSVL0023
      DELQ=QF*(GB-GA)*((BL-AL)/(AL*BL)+(AC-ADOTB)*Q/GA)/CLS             QSVL0024
35    QI=QI+DELQ*AXBI                                                   QSVL0025
      QJ=QJ+DELQ*AXBJ                                                   QSVL0026
      QK=QK+DELQ*AXBK                                                   QSVL0027
40    CONTINUE                                                          QSVL0028
      SQ=ABS(QI)+ABS(QJ)+ABS(QK)                                        QSVL0029
      IF (SQ.GT.QMO)   WRITE (6,100) I,J,L,KPHI,KP,KT,K,MGO,KDLL,LL,LD, QSVL0030
     1 XA,YA,ZA,XB,YB,ZB,QI,QJ,QK,SQ                                    QSVL0031
100   FORMAT ('0IJLKPHI=',I1,I3,I2,I3,' KPKTKMG=',I2,2I3,I2,' KDLLLD=', QSVL0032
     1 I2,2I3,                                                          QSVL0033
     2 ' XYZ=',6F6.3,' QS=',F7.4,2F8.4,F9.4)                            QSVL0034
      RETURN                                                            QSVL0035
      END                                                               QSVL0036
```

```
      SUBROUTINE QCVL                                                   QCVL0001
C  WG 71                                                                QCVL0002
C  CURVED VORTEX LINE SEGMENT INDUCED VELOCITY                          QCVL0003
      COMMON XA,YA,ZA,XB,YB,ZB,ZM,DELL,GA,GB,RHOC,RHOS,QI,QJ,QK,        QCVL0004
     1 IEG,I,J,K,KP,KPHI,KT,KDLL,L,LD,LG,LL,MGO,NGO,ND,KPO,KSTRT,       QCVL0005
     2 QMO,QMB,SQ                                                       QCVL0006
      AXBI=YB*ZA-YA*ZB                                                  QCVL0007
      AXBJ=XA*ZB-XB*ZA                                                  QCVL0008
      AXBK=XB*YA-XA*YB                                                  QCVL0009
      AXBS=AXBI**2+AXBJ**2+AXBK**2                                      QCVL0010
      AS=XA**2+YA**2+ZA**2                                              QCVL0011
      BS=XB**2+YB**2+ZB**2                                              QCVL0012
      AB=XA*XB+YA*YB+ZA*ZB                                              QCVL0013
      A=SQRT(AS)                                                        QCVL0014
      B=SQRT(BS)                                                        QCVL0015
      AD=AS-AB                                                          QCVL0016
      BD=BS-AB                                                          QCVL0017
      ADS=AXBS+AD*AD                                                    QCVL0018
      BDS=AXBS+BD*BD                                                    QCVL0019
      BDR=SQRT(BDS)                                                     QCVL0020
      ADR=SQRT(ADS)                                                     QCVL0021
      RAB=RHOC*AXBS                                                     QCVL0022
      AL=4.0*A*(BDS-BD*BDR)/RAB                                         QCVL0023
      BL=4.0*B*(ADS-AD*ADR)/RAB                                         QCVL0024
      Q=(GA*(ALOG(AL)-0.25)+GB*(ALOG(BL)-0.25))/(A*BDR)                 QCVL0025
      QI=Q*AXBI                                                         QCVL0026
      QJ=Q*AXBJ                                                         QCVL0027
      QK=Q*AXBK                                                         QCVL0028
      SQ=ABS(QI)+ABS(QJ)+ABS(QK)                                        QCVL0029
      IF (SQ.GT.QMO)   WRITE (6,100) I,J,L,KPHI,KP,KT,K,MGO,KDLL,LL,LD, QCVL0030
     1 XA,YA,ZA,XB,YB,ZB,QI,QJ,QK,SQ                                    QCVL0031
  100 FORMAT ('0IJLKPHI=',I1,I3,I2,I3,' KPKTKMG=',I2,2I3,I2,' KDLLLD=', QCVL0032
     1 I2,2I3,                                                          QCVL0033
     2 ' XYZ=',6F6.3,' QC=',F7.4,2F8.4,F9.4)                            QCVL0034
      RETURN                                                            QCVL0035
      END                                                               QCVL0036
```

```
      SUBROUTINE QVS                                                    QVS 0001
C  WG 71                                                                QVS 0002
C  RECTANGULAR, PLANAR VORTEX SHEET INDUCED VELOCITY                    QVS 0003
      REAL I1,I2,I3,I4                                                  QVS 0004
      COMMON XA,YA,ZA,XB,YB,ZB,ZM,DELL,GA,GB,RHOC,RHOS,QI,QJ,QK,        QVS 0005
     1 IEG,I,J,K,KP,KPHI,KT,KDLL,L,LD,LG,LL,MGC,NGO,ND,KPO,KSTRT,       QVS 0006
     2 QMO,QMB,SQ                                                       QVS 0007
      DX=XB-XA                                                          QVS 0008
      DY=YB-YA                                                          QVS 0009
      DZ=ZB-ZA                                                          QVS 0010
      DXYS=DX*DX+DY*DY                                                  QVS 0011
      CLS=DXYS+DZ*DZ                                                    QVS 0012
      CL=SQRT(CLS)                                                      QVS 0013
      CA=-(XA*DX+YA*DY+ZA*DZ)/CL                                        QVS 0014
      CB=CL-CA                                                          QVS 0015
      ZI=(.5*(ZA+ZB)-ZM)/DELL                                           QVS 0016
      RAD=SQRT(DXYS-ZI*ZI*CLS)                                          QVS 0017
      ZID=ZI*DZ                                                         QVS 0018
      XI=(DY*RAD-DX*ZID)/DXYS                                           QVS 0019
      YI=(-DX*RAD-DY*ZID)/DXYS                                          QVS 0020
      XK=(YI*DZ-ZI*DY)/CL                                               QVS 0021
      YK=(ZI*DX-XI*DZ)/CL                                               QVS 0022
      ZK=(XI*DY-YI*DX)/CL                                               QVS 0023
      CAN=CA/CL                                                         QVS 0024
      XH=XA+DX*CAN                                                      QVS 0025
      YH=YA+DY*CAN                                                      QVS 0026
      ZH=ZA+DZ*CAN                                                      QVS 0027
      X1=XI*XH+YI*YH+ZI*ZH                                              QVS 0028
      Z=XK*XH+YK*YH+ZK*ZH                                               QVS 0029
      ZZ=Z*Z                                                            QVS 0030
      X2=X1-DELL                                                        QVS 0031
      X2S=X2*X2                                                         QVS 0032
      TDELL=0.01*DELL                                                   QVS 0033
      IF (X2S.LT.TDELL) GO TO 4                                         QVS 0034
      X1S=X1*X1                                                         QVS 0035
      IF (X1S.LT.TDELL) GO TO 5                                         QVS 0036
```

```
3     CONTINUE                                                          QVS 0037
      CAS=CA*CA                                                         QVS 0038
      CBS=CB*CB                                                         QVS 0039
      SOA=SQRT(X1S+ZZ+CAS)                                              QVS 0040
      SOB=SQRT(X1S+ZZ+CBS)                                              QVS 0041
      STA=SQRT(X2S+ZZ+CAS)                                              QVS 0042
      STB=SQRT(X2S+ZZ+CBS)                                              QVS 0043
      I1=(GA/DELL)*(ATAN(CA*X2/(Z*STA))-ATAN(CA*X1/(Z*SOA))            QVS 0044
     1 +ATAN(CB*X2/(Z*STB))-ATAN(CB*X1/(Z*SOB)))                        QVS 0045
      I2=GA*ALOG(((STA+CA)*(SOA-CA)*(STB+CB)*(SOB-CB))/                 QVS 0046
     1 ((STA-CA)*(SOA+CA)*(STB-CB)*(SOB+CB)))/(2.0*DELL)                QVS 0047
      QI=I1*XI+I2*XK                                                    QVS 0048
      QJ=I1*YI+I2*YK                                                    QVS 0049
      QK=I1*ZI+I2*ZK                                                    QVS 0050
C   IEG = 1 FOR GA.NE.GB; IEG = 2 FOR GA=GB                             QVS 0051
      GO TO (7,8),IEG                                                   QVS 0052
4     X2S=TDELL                                                         QVS 0053
      X2=SIGN(0.1*DELL,X2)                                              QVS 0054
      X1=X2+DELL                                                        QVS 0055
      X1S=X1*X1                                                         QVS 0056
      GO TO 3                                                           QVS 0057
5     X1S=TDELL                                                         QVS 0058
      X1=SIGN(0.1*DELL,X1)                                              QVS 0059
      X2=X1-DELL                                                        QVS 0060
      X2S=X2*X2                                                         QVS 0061
      GO TO 3                                                           QVS 0062
7     CONTINUE                                                          QVS 0063
      G=(GB-GA)/CL                                                      QVS 0064
      I3=Z*ALOG(((STA+X2)*(SOB+X1))/((SOA+X1)*(STB+X2)))/DELL           QVS 0065
      I4=(SOA-STA+STB-SOB)/DELL                                         QVS 0066
      CG=CA/GA                                                          QVS 0067
      QI=QI+G*(CG*QI+I3*XI+I4*XK)                                       QVS 0068
      QJ=QJ+G*(CG*QJ+I3*YI+I4*YK)                                       QVS 0069
      QK=QK+G*(CG*QK+I3*ZI+I4*ZK)                                       QVS 0070
8     CONTINUE                                                          QVS 0071
      SQ=ABS(QI)+ABS(QJ)+ABS(QK)                                        QVS 0072
```

```
      IF (SQ.GT.QMO)   WRITE (6,100) I,J,L,KPHI,KP,KT,K,MGD,KDLL,LL,LD,        QVS 0073
     1 XA,YA,ZA,XB,YB,ZB,QI,QJ,QK,SQ                                            QVS 0074
  100 FORMAT ('0IJLKPHI=',I1,I3,I2,I3,' KPKTKMG=',I2,2I3,I2,' KDLLLD=',       QVS 0075
     1 I2,2I3,                                                                  QVS 0076
     2 ' XYZ=',6F6.3,' QA=',F7.4,2F8.4,F9.4)                                    QVS 0077
      RETURN                                                                    QVS 0078
      END                                                                       QVS 0079
```

```
      SUBROUTINE SPLOT (NPHI,NPSI,TAD,DANG)
      DIMENSION B(512),SX(99),SY(99),SZ(99),ABEL(15),BBEL(15),ZLAM(99),
     1 OSX(99),ZZK(97,25),TX(15),TY(15),TZ(15)
      COMMON/SSPLOT/EMU,AMBDA,ETA,C(97),S(97),PSI(97),D(3,25,97)
      NAMELIST/DPLT/ILAB,SR,SK
      NIN=5
C  READ IN SCALE FACTORS
      READ (NIN,DPLT)
      IF (ILAB.EQ.0) GO TO 10
C  READ PLOT LABEL
      READ (NIN,100) (ABEL(I),I=1,15)
      READ (NIN,100) (BBEL(I),I=1,15)
100   FORMAT (15A4)
10    CONTINUF
      DR=1.0/SR
      DZ=1.0/SK
      TDR=-2.0*DR
      SDR=-7.0*DR
      TDZ=-3.0*DZ
      NPO=NPHI+1
      NPT=NPHI+2
C  PUT FIRSTV AND DELTA INTO ARRAYS TO BE PLOTTED
      SX(NPO)=0.0
      SX(NPT)=DR
      SY(NPO)=SDR
      SY(NPT)=DR
      ZLAM(NPO)=TDZ
      ZLAM(NPT)=DZ
C  RIGID WAKE Z COMPONENT
      DO 15 K=1,NPHI
15    ZLAM(K)=-AMBDA*PSI(K)
C  DISTORTED WAKE Z COMPONENT
      DO 20 J=1,NPSI
      ADD=0.0
      DO 20 K=1,NPHI
      ZZK(K,J)=ADD-D(3,J,K)
```

```
20    ADD=ADD+TAD                                                       SPLT0037
      DEMU=EMU*DANG                                                     SPLT0038
C  READ IN STARTING PSI                                                 SPLT0039
30    READ (NIN,101) J1,NS                                              SPLT0040
101   FORMAT (I3,I2)                                                    SPLT0041
      PSIS=DANG*FLOAT(J1-1)                                             SPLT0042
C  COMPUTE RIGID WAKE TPP VIEW                                          SPLT0043
      ADD=1.0                                                           SPLT0044
      J=J1                                                              SPLT0045
      DO 31 K=1,NPHI                                                    SPLT0046
      SX(K)=ADD+C(J)                                                    SPLT0047
      SY(K)=S(J)                                                        SPLT0048
      ADD=ADD+DEMU                                                      SPLT0049
      J=J-1                                                             SPLT0050
      IF (J.EQ.0) J=NPSI-1                                              SPLT0051
31    CONTINUE                                                          SPLT0052
C  LENGTH OF PLOT AXIS                                                  SPLT0053
      MM=SR*(2.0+DEMU*NPHI)                                             SPLT0054
      IF (MM.LT.8) MM=8                                                 SPLT0055
      XMM=MM                                                            SPLT0056
      DMM=XMM+2.0                                                       SPLT0057
      IF (ILAB.EQ.0) GO TO 33                                           SPLT0058
C  PLOT LABEL PREVIOUSLY READ IN                                        SPLT0059
      CALL SYMBOL (0.1,10.,.14,ABEL,0.0,60)                             SPLT0060
      CALL SYMBOL (0.1,9.8,.14,BBEL,0.0,60)                             SPLT0061
33    CONTINUE                                                          SPLT0062
C  PLOT LINE OF HUB MOTION                                              SPLT0063
      CALL PLOT (0.0,7.0,3)                                             SPLT0064
      CALL PLOT (XMM,7.0,2)                                             SPLT0065
C  PLOT TIP PATH PLANE (TPP)                                            SPLT0066
      CALL PLOT (XMM,3.0,3)                                             SPLT0067
      CALL PLOT (0.0,3.0,2)                                             SPLT0068
C  PLOT HUB AXIS                                                        SPLT0069
      CALL PLOT (2.0,0.5,3)                                             SPLT0070
      CALL PLOT (2.0,9.0,2)                                             SPLT0071
C  X AXIS = CALCOMP Y AXIS                                              SPLT0072
```

```
      CALL AXIS (0.0,5.0,'X',1,4.0,90.0,TDR,DR)                         SPLT0073
C  Y AXIS = CALCOMP X AXIS                                              SPLT0074
      CALL AXIS (0.0,4.0,'Y',1,XMM,0.0,0.0,DR)                          SPLT0075
C  PLOT RIGID WAKE TPP VIEW                                             SPLT0076
      CALL LINE (SX,SY,NPHI,1,0,4)                                      SPLT0077
C  Z AXIS = CALCOMP -Y AXIS                                             SPLT0078
      CALL AXIS (0.0,3.0,'Z',-1,3.0,270.0,0.0,DZ)                       SPLT0079
C  PLOT RIGID WAKE SIDE VIEW                                            SPLT0080
      CALL LINE (SX,ZLAM,NPHI,1,0,4)                                    SPLT0081
      JS=J1                                                             SPLT0082
C  DISTORTED WAKE IS PLOTTED WITH TWO DIFFERENT SYMBOLS - ISYM=3,4      SPLT0083
C  ISYM=3 IS FOR PSI = 0 - 180,  ISYM=4 IS FOR PSI = 180 - 360          SPLT0084
      NPM=(NPSI-1)/2+1                                                  SPLT0085
      JP=1                                                              SPLT0086
      ISYM=3                                                            SPLT0087
      NP=JS                                                             SPLT0088
      IF (NP.LE.NPM) GO TO 35                                           SPLT0089
      NP=JS-NPM+1                                                       SPLT0090
      ISYM=4                                                            SPLT0091
35    DO 40 J=1,NP                                                      SPLT0092
      TX(J)=SX(JP)+D(2,JS,JP)                                           SPLT0093
      TY(J)=SY(JP)+D(1,JS,JP)                                           SPLT0094
      TZ(J)=ZZK(JP,JS)                                                  SPLT0095
      JP=JP+1                                                           SPLT0096
      JS=JS-1                                                           SPLT0097
      IF (JS.EQ.0) JS=NPSI-1                                            SPLT0098
40    CONTINUE                                                          SPLT0099
      NPO=NP+1                                                          SPLT0100
      NPT=NP+2                                                          SPLT0101
C  PUT FIRSTV AND DELTA INTO ARRAYS TO BE PLOTTED                       SPLT0102
      TX(NPO)=0.0                                                       SPLT0103
      TX(NPT)=DR                                                        SPLT0104
      TY(NPO)=SDR                                                       SPLT0105
      TY(NPT)=DR                                                        SPLT0106
      TZ(NPO)=TDZ                                                       SPLT0107
      TZ(NPT)=DZ                                                        SPLT0108
```

```
C    PLOT DISTORTED WAKE TPP VIEW                                       SPLT0109
      CALL LINE (TX,TY,NP,1,1,ISYM)                                     SPLT0110
C    PLOT DISTORTED WAKE SIDE VIEW                                      SPLT0111
      CALL LINE (TX,TZ,NP,1,1,ISYM)                                     SPLT0112
      IF (JP.GT.NPHI) GO TO 50                                          SPLT0113
      JP=JP-1                                                           SPLT0114
      JS=JS+1                                                           SPLT0115
      NP=NPHI-JP+1                                                      SPLT0116
      IF (NP.GT.NPM) NP=NPM                                             SPLT0117
      IF (ISYM.EQ.4) GO TO 45                                           SPLT0118
      ISYM=4                                                            SPLT0119
      GO TO 35                                                          SPLT0120
45    ISYM=3                                                            SPLT0121
      GO TO 35                                                          SPLT0122
50    CONTINUE                                                          SPLT0123
      CALL PLOT (DMM,0.0,-3)                                            SPLT0124
      NS=NS-1                                                           SPLT0125
      IF (NS.GT.0) GO TO 30                                             SPLT0126
      RETURN                                                            SPLT0127
      END                                                               SPLT0128
```

```
      SUBROUTINE FXOF (M,N,DATA)                                        FXOF0001
C  WG 71                                                                FXOF0002
C  PRINT DUMP IF ERROR                                                  FXOF0003
      INTEGER OKPHI,SKPHI                                               FXOF0004
      COMMON XA,YA,ZA,XB,YB,ZB,ZM,DELL,GA,GB,RHOC,RHOS,QI,QJ,QK,        FXOF0005
     1 IEG,I,J,K,KP,KPHI,KT,KDLL,L,LD,LG,LL,MGO,NGO,ND,KPO,KSTRT,       FXOF0006
     2 QMB,QMO,SQ                                                       FXOF0007
      COMMON/SSPLOT/EMU,AMBDA,ETA,C(97),S(97),PSI(97),D(3,25,97)        FXOF0008
      COMMON/SGAM/IABC,I1,IG,IS,LST,NL,NB,NPSI,NPHI,NIN,NOUT,NDI,       FXOF0009
     1 EM,DPSI,DANG,TAMB,GZER,BOER,                                     FXOF0010
     2 EL(6),ELB(25),ANG(97),ZETA(6),GBV(6,25),BVG(6,25),GS(6,25),      FXOF0011
     3 ENDM(25),NDM(25)                                                 FXOF0012
      COMMON/SQCAL/OKPHI,SKPHI,OLDM,PHI,SUMI,SUMJ,SUMK,                 FXOF0013
     1 OSUMI,OSUMJ,OSUMK,TANMU,TAMD,PZA,PZAP,AD,ETASP,ETACP,DELTA,DMS,  FXOF0014
     2 RHOK,RHOG,START,NBIGM,                                           FXOF0015
     1 AD(3),D3(3),VI(6),VJ(6),VK(6),OVI(6),OVJ(6),OVK(6),              FXOF0016
     2 DVO(3,25),DS(3,25),DSN(3,25),TOT(3,25),CUR(3,25),QBV(3),         FXOF0017
     3 KM(6,25),KTR(6,25,16),VEL(3,25,97),KPB(25)                       FXOF0018
      COMMON/SDCAL/IPR,NPSR,LDM,FWD,EWD,DSMI,DSMJ,DSMK,DSMS             FXOF0019
      NAMELIST/DUMP/I,J,K,KP,KPHI,OKPHI,SKPHI,KSTRT,KT,KDLL,            FXOF0020
     1 L,LG,LD,LL,LST,MGO,NGO,IABC,I1,IS,IG,IEG,NL,NB,NPSI,NPHI,NDI,    FXOF0021
     2 EM,DPSI,DANG,EMU,AMBDA,TANMU,TAMB,TAMD,PHI,PZA,PZAP,             FXOF0022
     3 ETA,ETACP,ETASP,AD,BOER,GZER,DELTA,DMS,OLDM,RHOK,RHOG,RHOC,      FXOF0023
     4 SUMI,SUMJ,SUMK,OSUMI,OSUMJ,OSUMK,QI,QJ,QK,SQ,QMB,QMO,            FXOF0024
     5 XA,YA,ZA,XB,YB,ZB,ZM,DELL,GA,GB,DSMI,DSMJ,DSMK,DSMS,FWD,EWD,NPSR, FXOF0025
     6 AD,D3,VI,VJ,VK,OVI,OVJ,OVK,ZETA,PSI,EL,ELB,NDM,KPB,GBV,BVG,GS,   FXOF0026
     7 KM,QBV,TOT,CUR,VEL,KTR,DVO,DS,DSN,D                              FXOF0027
      WRITE (6,DUMP)                                                    FXOF0028
      STOP                                                              FXOF0029
      END                                                               FXOF0030
```

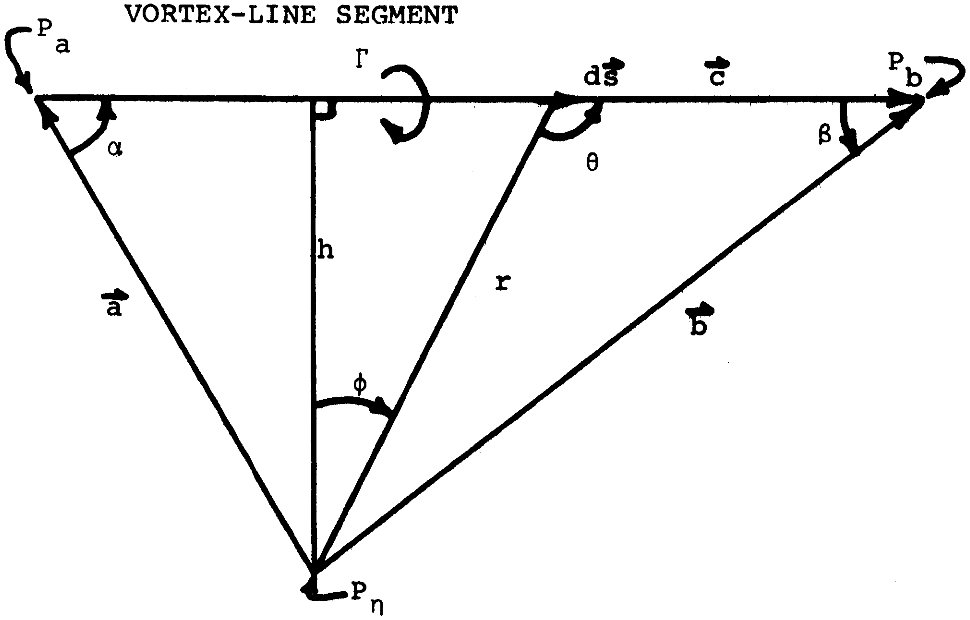

FIG. A.1 VORTEX-LINE SEGMENT GEOMETRY

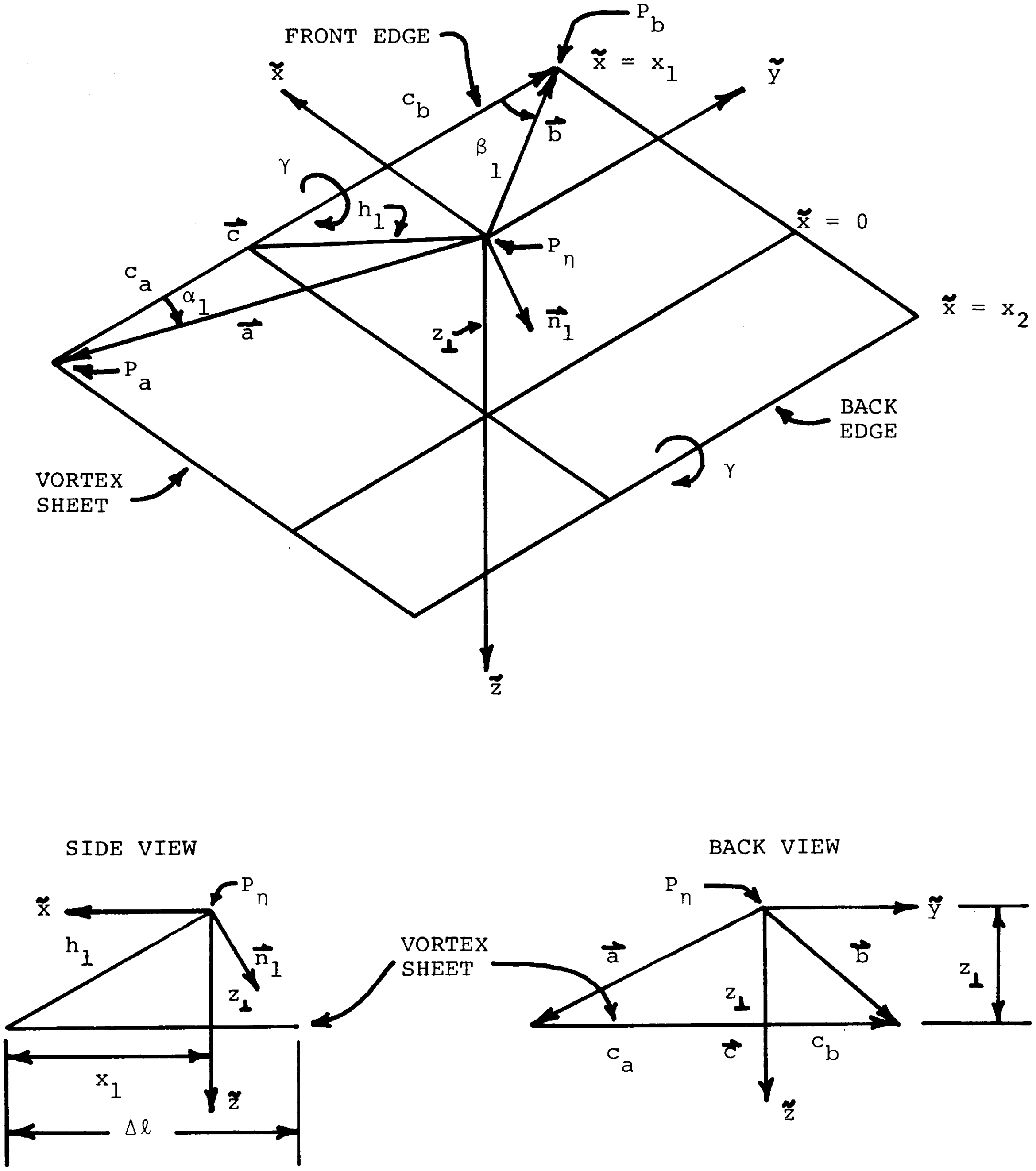

FIG. B.1 VORTEX-SHEET GEOMETRY AND COORDINATE SYSTEM

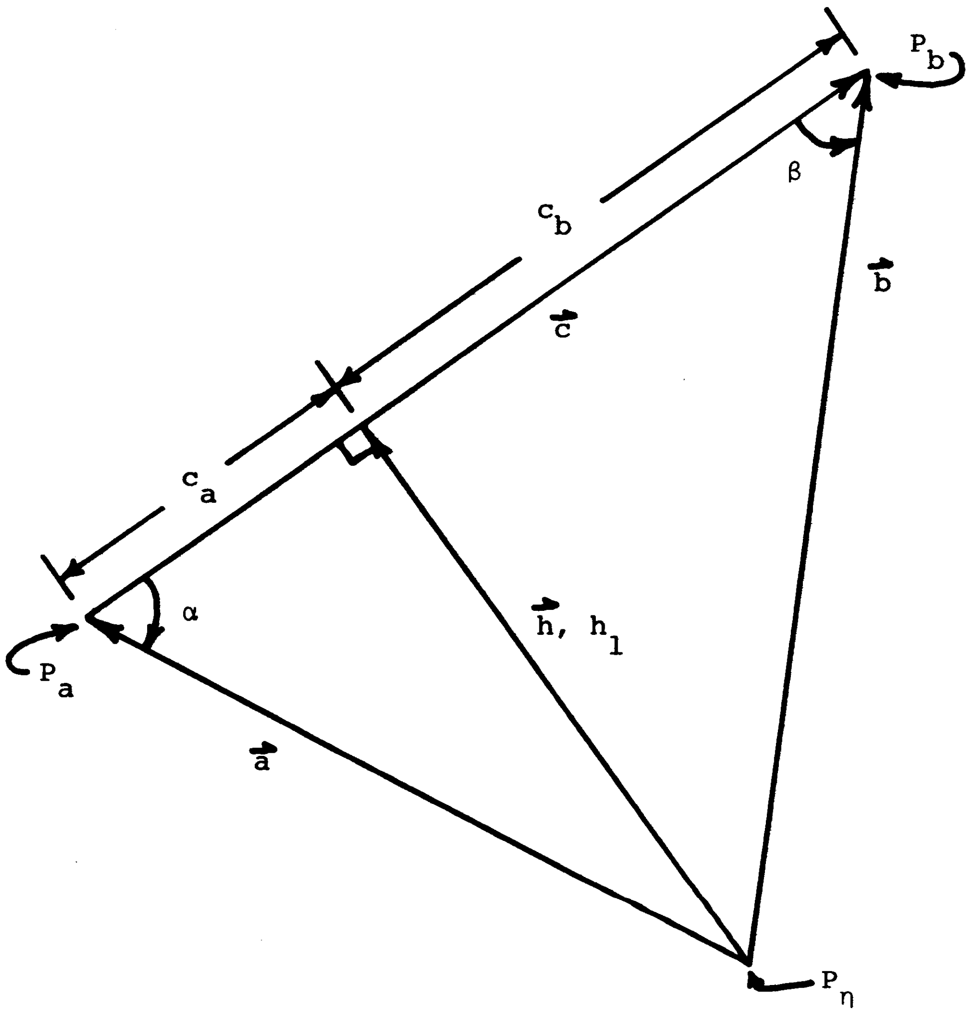

FIG. B.2 VIEW IN $\vec{a}$, $\vec{b}$ PLANE

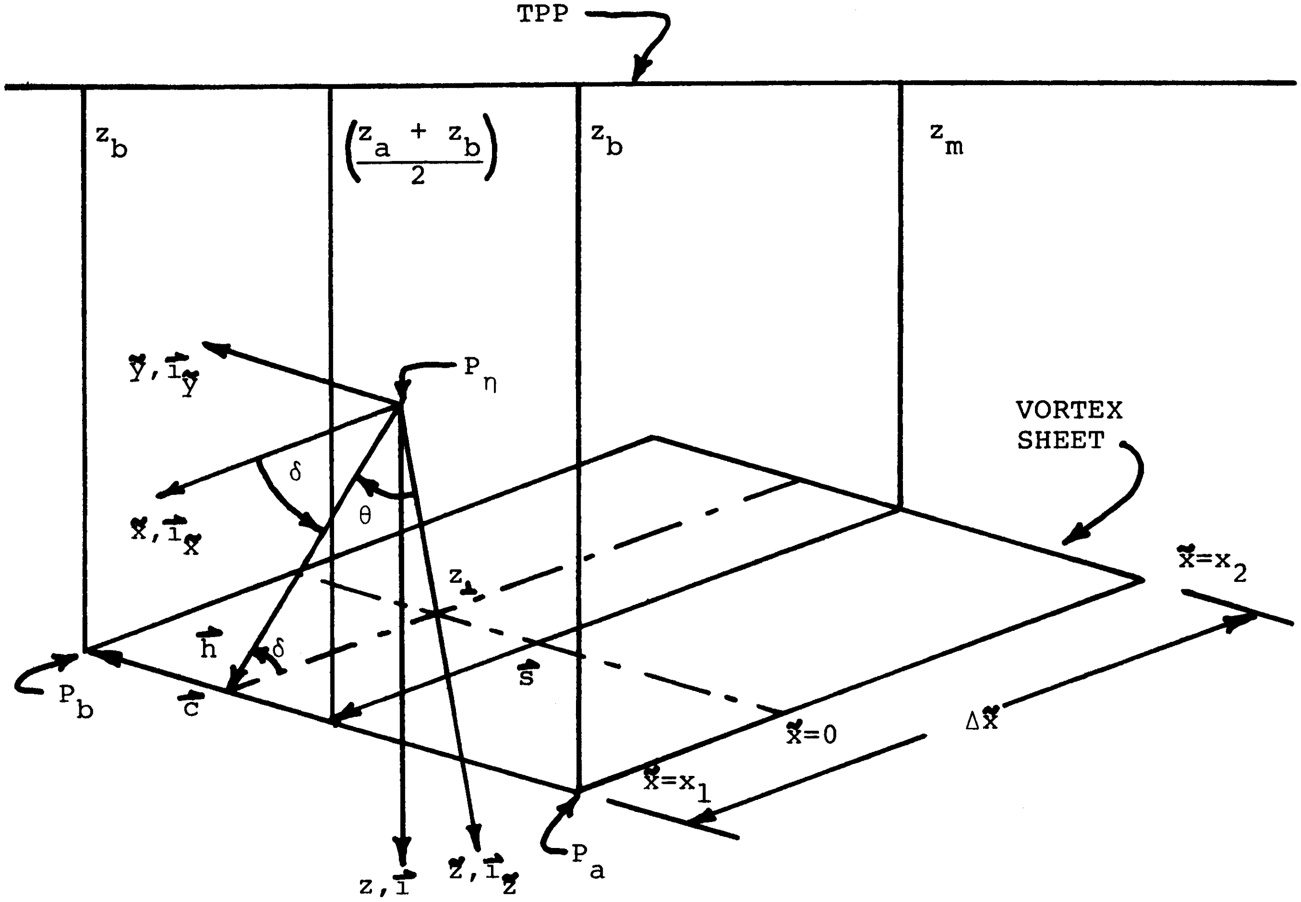

FIG. B.3 VIEW OF VORTEX SHEET VIEWED PARALLEL TO TPP

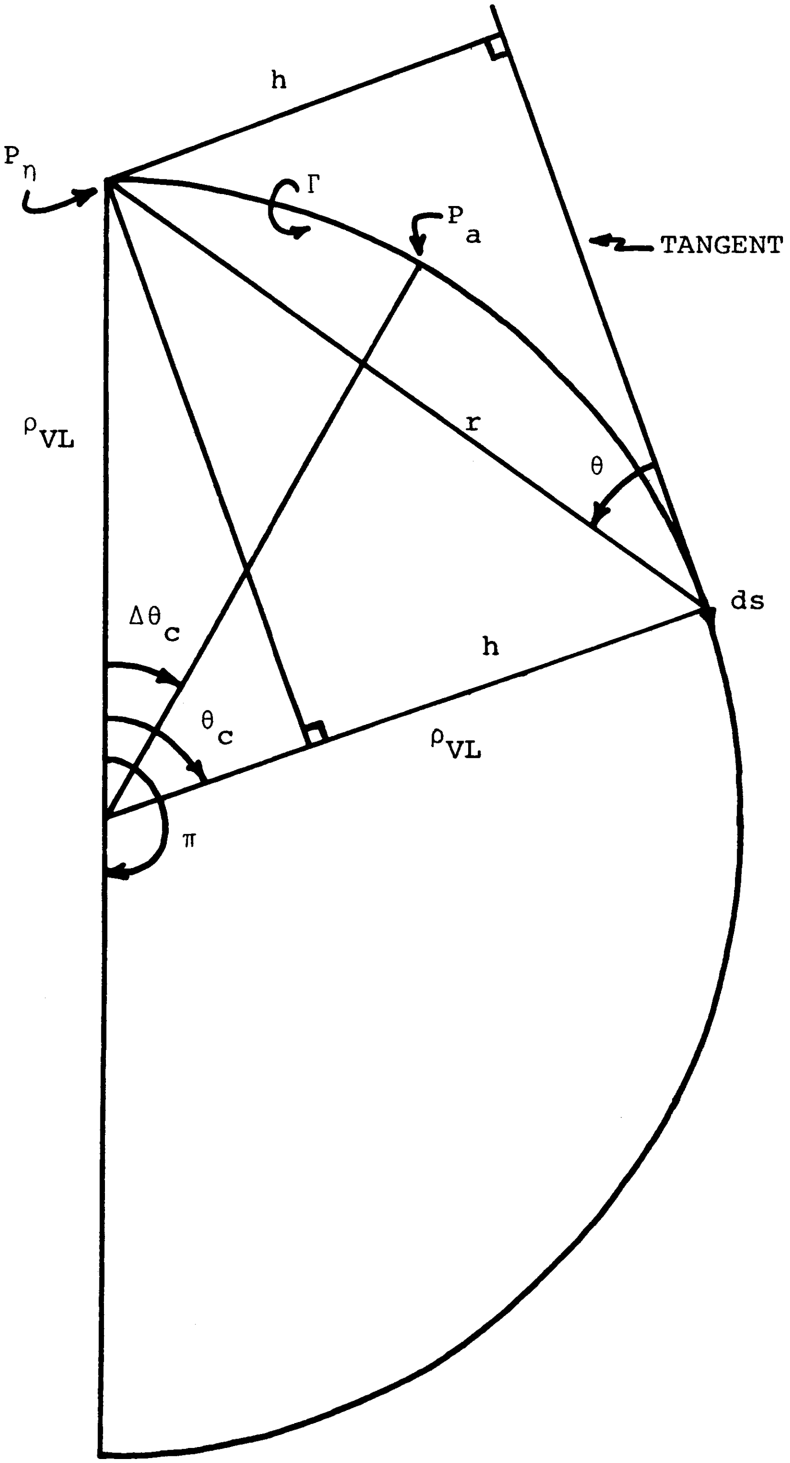

FIG. C.1 CURVED VORTEX LINE SEGMENT AS PART OF A HALF VORTEX RING

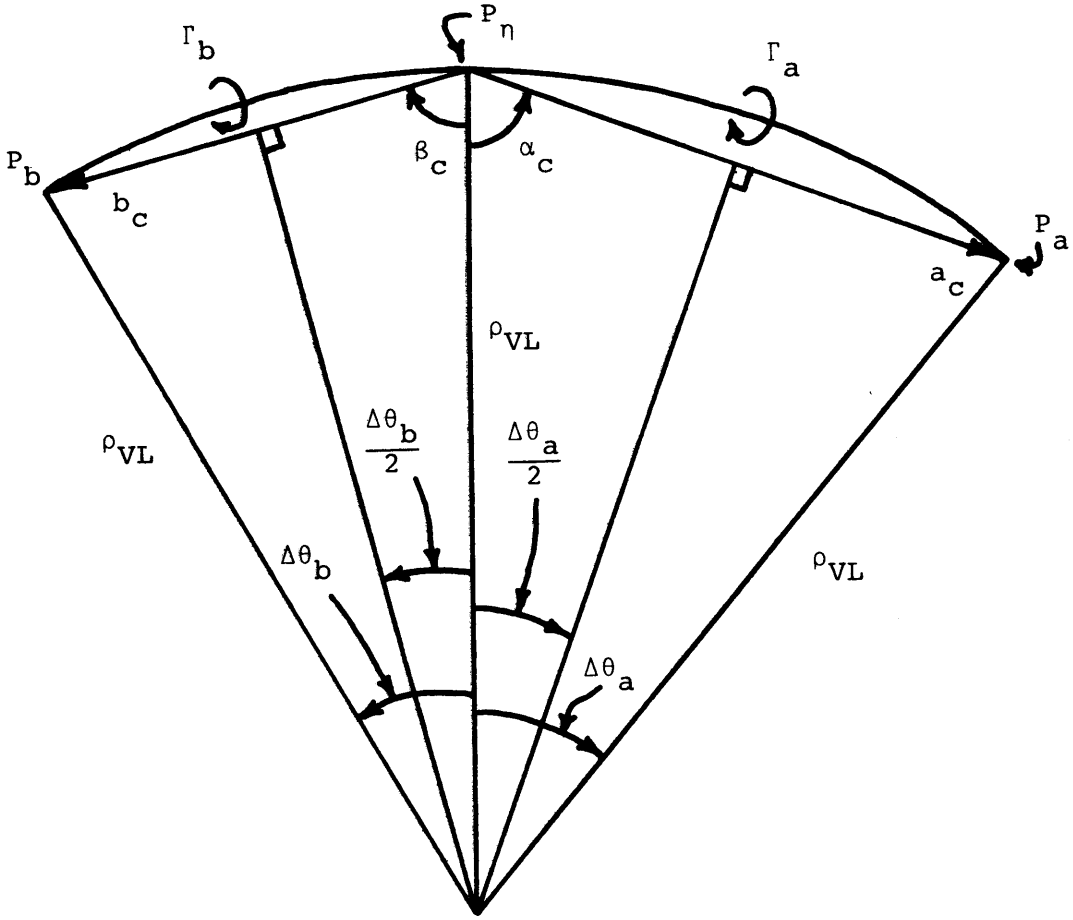

FIG. C.2 CURVED VORTEX-LINE-SEGMENT GEOMETRY

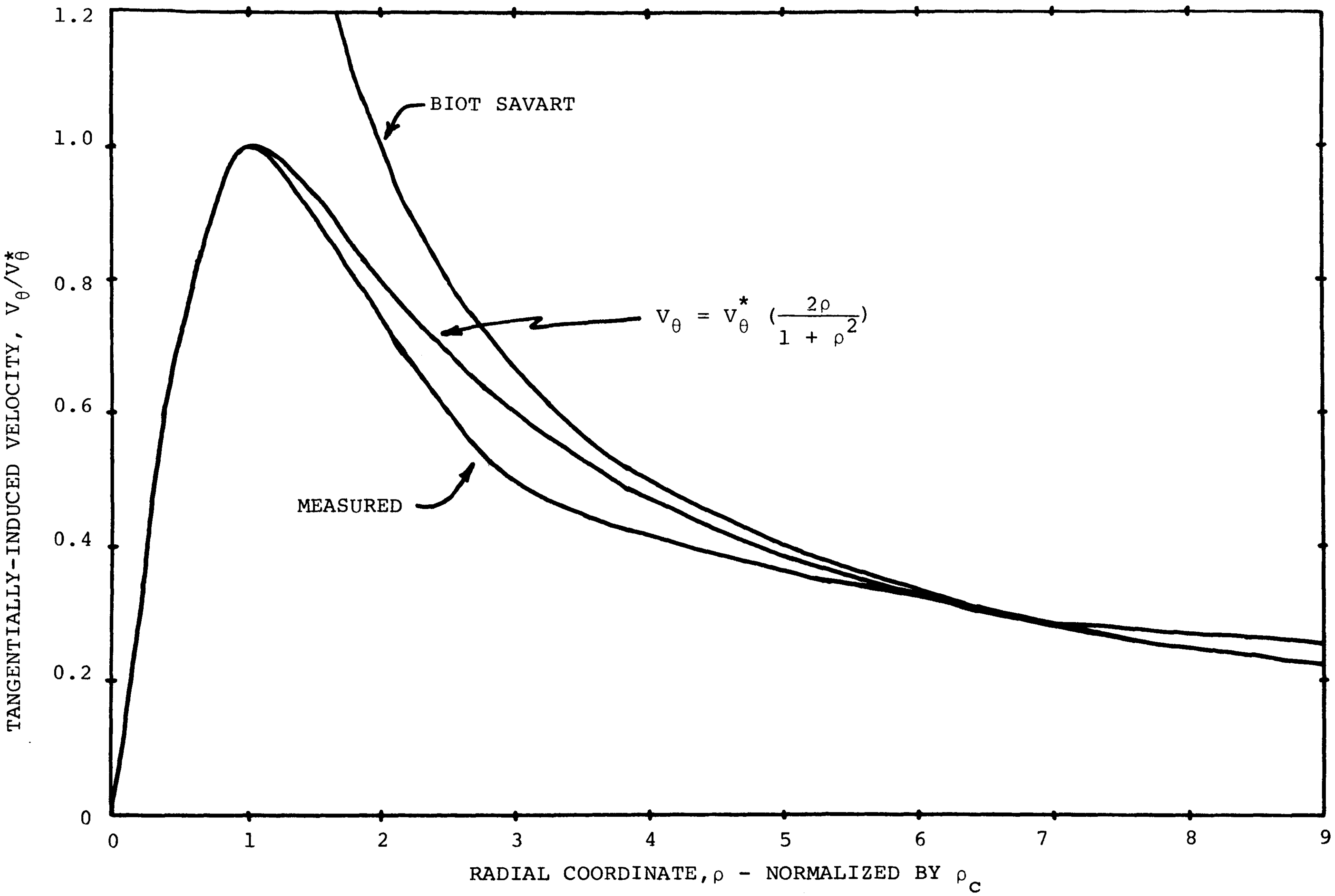

FIG. D.1 COMPARISON OF FWVCM WITH MEASURED DATA AND BIOT SAVART

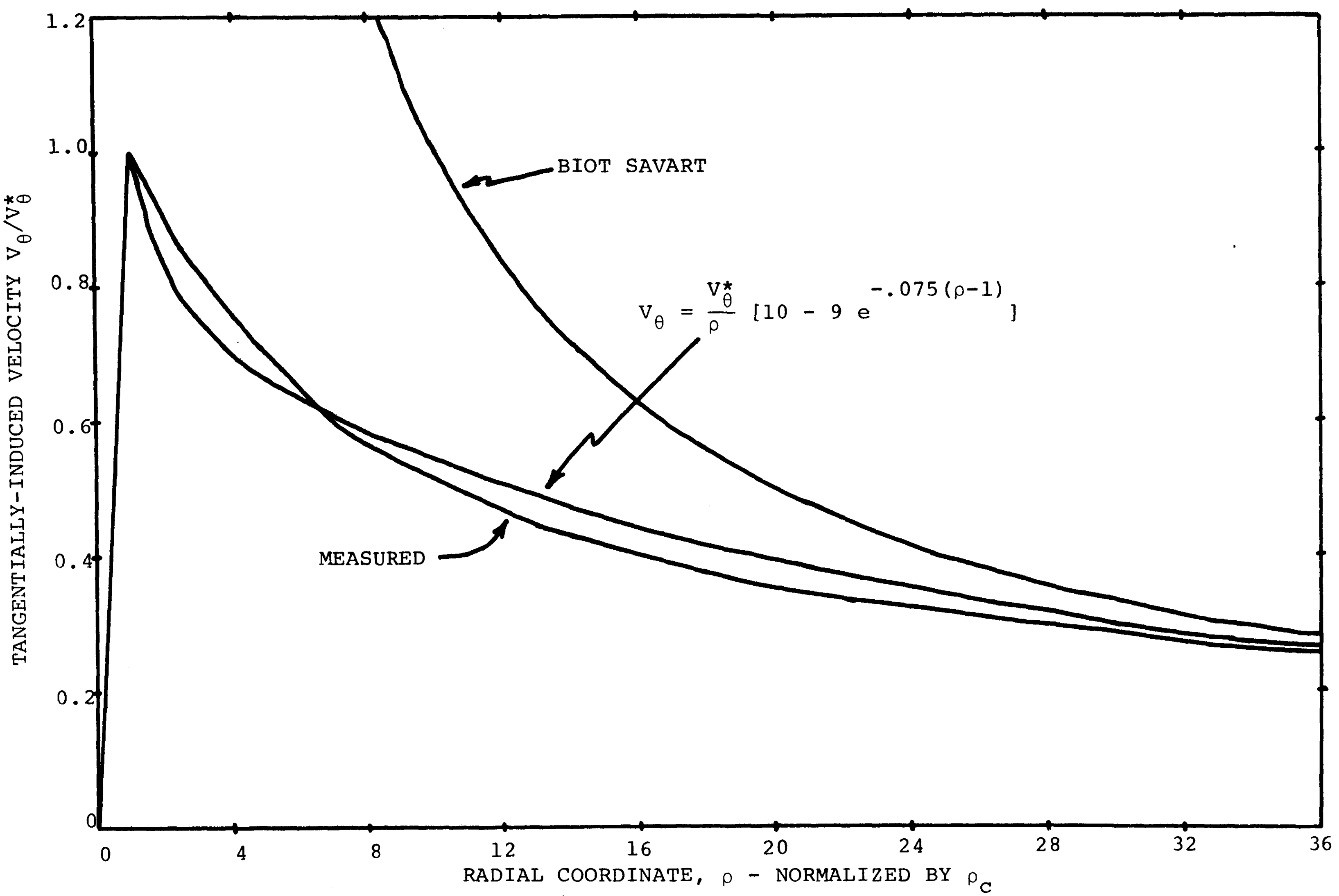

FIG. D.2 COMPARISON OF RWVCM WITH MEASURED DATA AND BIOT SAVART

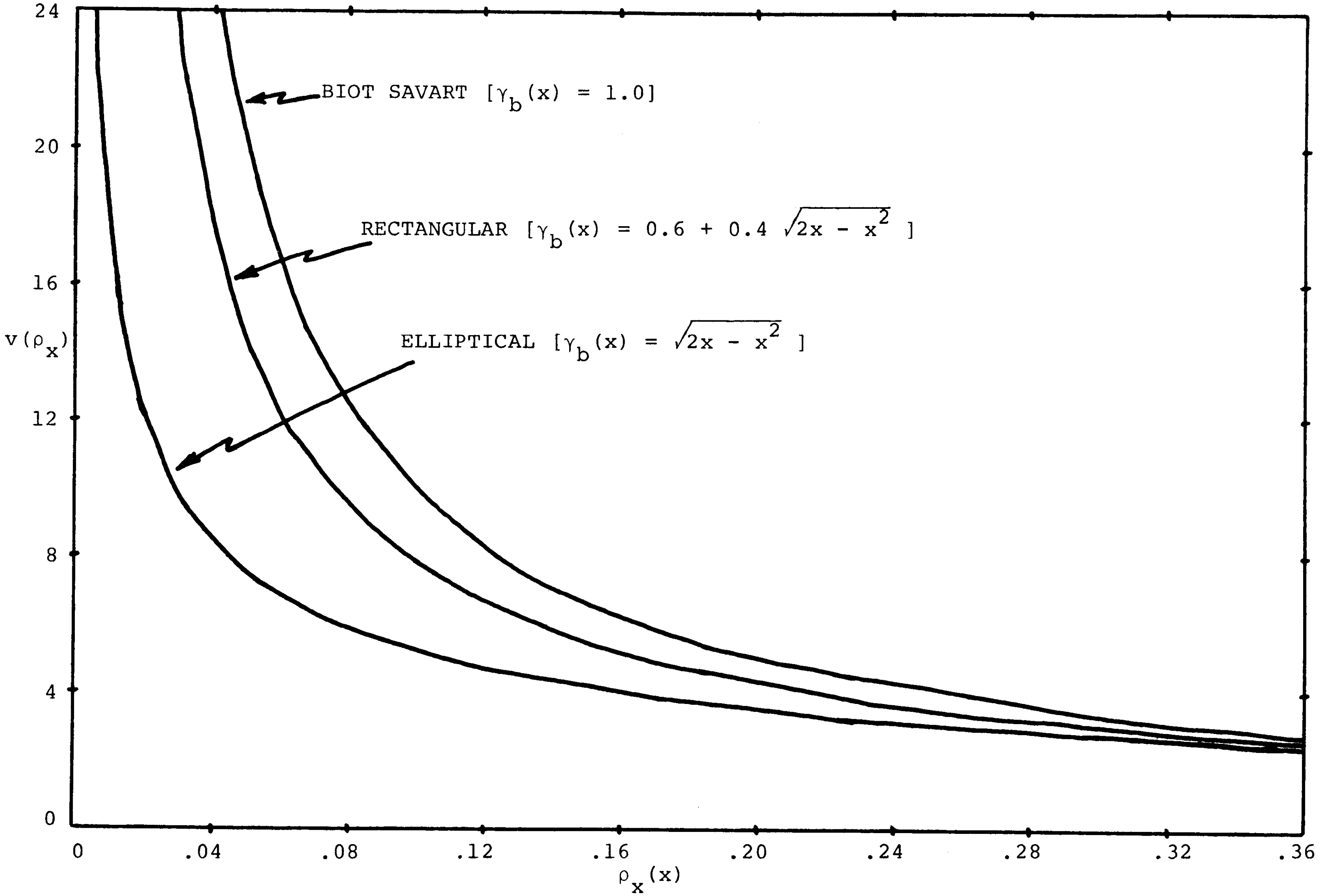

FIG. D.3 EFFECT OF $\gamma_b(x)$ ON $v(\rho_x)$ VS. $\rho_x(x)$

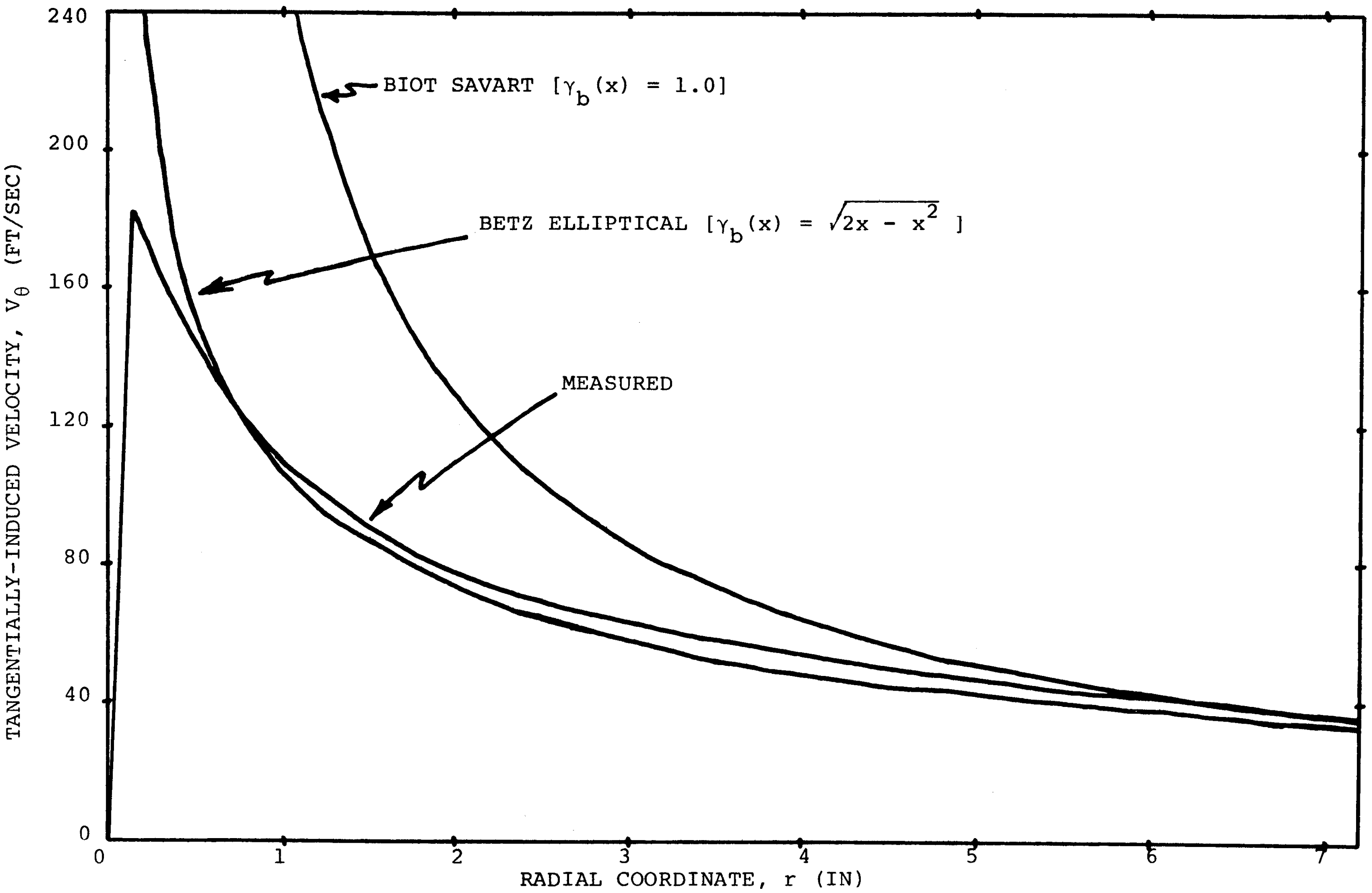

FIG. D.4 BETZ MODEL VS. MEASURED DATA - ROTARY WING [39]

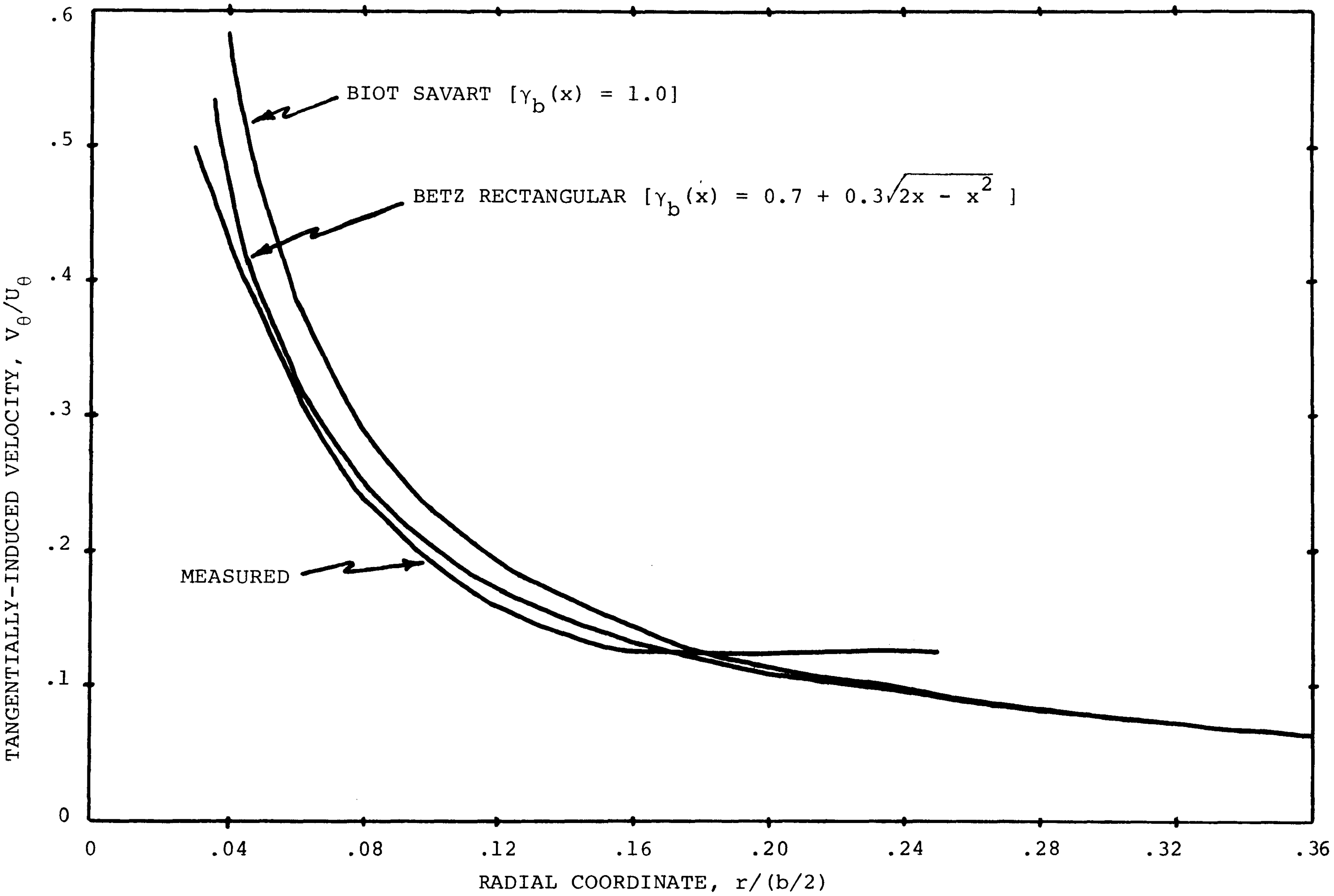

FIG. D.5 BETZ MODEL VS. MEASURED DATA - FIXED WING [43]

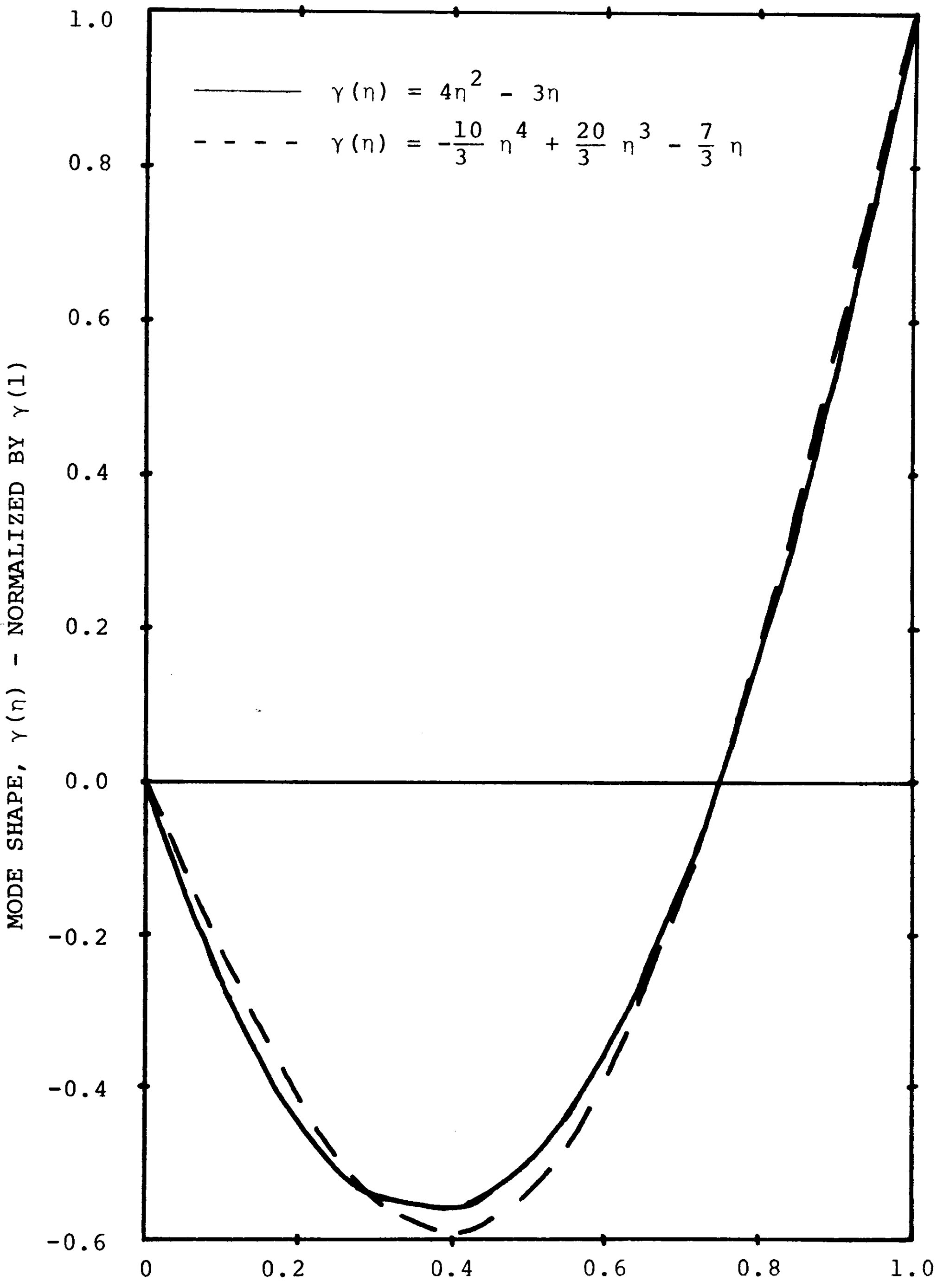

FIG. G.1 COMPARISON OF QUADRATIC VS. QUARTIC BLADE BENDING MODE SHAPES

BIOGRAPHY

Michael P. Scully was born on November 17, 1942 in Albany, California. After graduation from Pleasant Hill High School in Pleasant Hill, California he entered M.I.T. in 1960. He received a Bachelor of Science Degree from the Department of Aeronautics and Astronautics in June 1964. A four-month CO-OP work period was spent at Boeing, Seattle in 1962/3. Entering graduate school at the Massachusetts Institute of Technology, Department of Aeronautics and Astronautics in September 1964, he received a Master of Science Degree and an Engineer in Aeronautics and Astronautics Degree in June 1967. He entered the doctoral program at the M.I.T. Department of Aeronautics and Astronautics in September 1967. His graduate education has been supported by a full-time Research Assistantship with both the Aeroelastic and Structures Research Laboratory and the Flight Transportation Laboratory. In addition, there were two summer DSR staff appointments and a short period on National Science Foundation Traineeship.